A. M. Turing Award Oral History Interview with Manuel Blum

by Ann Gibbons

Pittsburgh, Pennsylvania

Oct. 26, 2017

Gibbons: I'm Ann Gibbons. This is Manuel Blum, and I'm interviewing him in his home in Pittsburgh near Carnegie Mellon University. Today is Thursday, the 26th of October, 2017.

So Manuel, we're going to start with your childhood, your background … get some information about your family, your beginnings. Tell me where you were born, where you grew up, and a little bit about your family.

Blum: Okay, good. I was born in Caracas, Venezuela. My parents were from Romania. They spoke Romanian in the schools, they spoke German at home, and my first language actually was German, even though I was in Venezuela where people speak Spanish. It's very fortunate that they got to Venezuela. They tried to get into the US but couldn't. It was hard to get a visa. They didn't have much money and the Sephardic Jews in Venezuela put up the money for the Ashkenazi Jews to come in. That's how they were able to get into Venezuela.

Gibbons: Had they faced persecution, or did they just see the writing on the wall in Europe? Why did they leave then?

Blum: I think they left because they were hungry. They just didn't have enough food to eat. My father really wanted to have an education, but he was taken out of school in the sixth grade because they needed him to work. This was pretty sad since he wanted to be a doctor. They came to Venezuela. They had me. [laughs] So I have memories from that. It's sort of fortunate that we moved from one place to another. Every year, we'd move someplace, so I have my memories from each place. If people move, then you remember where that memory is from.

My first memory actually is when I was less than a year old. You might wonder how come I remember it. It's because it was important to me to be able to have these memories. The memory from when I was less than one, I know it was less than one year old because my mother was pregnant with my brother, who's one year younger. I remember we were walking along a trail, grass, a brook, and trees that went up to the sky. That's my memory of it. I mentioned it to my mother just much later. When I was already a professor, I mentioned to her that I had this memory, and she said, "Oh yeah." On that trail, she saw a snake, and it scared the hell out of her and she picked me up and ran back. I don't know any of that. But the interesting thing for me about that memory is that I'm not in it. All my memories, I see myself in it. They clearly have been worked over. This was a memory where no, I'm not in it. I see the brook, I see the trees. I don't see myself.

Then there are other memories. I have a memory from when I was three years old. We left Venezuela when I was four. The memory is of my father with a chicken in one hand and a razor blade in the other, [laughs] and I was praying so hard for the chicken. Then my father lost the grip on the chicken and the chicken went flying into the trees and I was *so* grateful. I'm pretty sure… I never asked him about this, but I suspect that he took one look at me and saw that I was about to keel over, [laughs] and he decided to the let the chicken go.

Gibbons: So you moved to the United States when you were four.

Blum: Four, right.

Gibbons: Tell me about that move. You moved to New York, or to Miami you said?

Blum: First to Miami. I was four years old. It's 1942. The airplane I took was a biplane seaplane. To get to Miami, it had to stop at Cuba on the way to fill up with gas. I remember that plane trip. It was very cool. When the airplane landed, the water came up and I thought, "Oh, we're going to go underwater." It was great.

All the time, I was speaking German with my parents. We got to this country and people didn't like to hear German spoken. We were at war. This country was at war with Germany. They did *not* want to hear German spoken, so they decided, hmm, they don't speak English, they'll teach me Spanish. [laughs] And in fact I learned Spanish. I remember the shocked look on my mother's face about a month after I had started with Spanish when she realized that I'd completely forgotten German. She points to a window and says, "Don't you remember how to say 'window' in German? *Fenster*." No, I did not remember *Fenster*. I did not know I even knew *Fenster*.

So all my early years, I never spoke the language of the people I was with. In fact, I learned English, I guess I began learning when I was about to go into kindergarten, because I knew that I would have to learn English. So I went to a corner where I saw two women talking to each other and I turned my back to them so they wouldn't notice that I'm eavesdropping, and I listened. I remember telling myself exactly, "I will never learn this language. They speak much too fast for me." The wonderful thing about this is that I remember saying it to myself in English. [laughs] Of course, I didn't know any English, so it's an interesting way that we… an interesting thing about our memories. We now know that you pull out this memory and then you put it back and it's a little changed.

Gibbons: It's also interesting that language is a key to how brains are developing and forming as well. Your memories seem to pick up on that, your interest in the brain. The memories that are seared in by fear – a snake, perhaps.

Blum: No, I don't remember the snake. It was… Okay. I don't remember the snake. That was my mother.

Gibbons: What did your parents do? Why did they move from Venezuela to Miami and then New York?

Blum: Well, part of it was that I had amoebic dysentery. They didn't have good medicine there, so they came up to get me into hospital. I had amoebic dysentery because we had a little pool in the back the size of a bathtub and my father would tell me, "Don't drink the water," but I loved to have my mouth at water level and to let the water come in. [laughs] This was not good. I can imagine looking at that water under a microscope. It would be full of amoebas going back and forth.

Let's see, what else?

Gibbons: So you came because you were sick. Did your father find work in Florida first and then in New York?

Blum: Yeah.

Gibbons: What did he do?

Blum: He was a watchmaker. In fact, his exam required him to make a watch from scratch. He had to design it, cut out all the pieces, the gears, and have a functioning watch out of that. I was very proud of him for that. And when he went to Venezuela, he made a name for himself because there was a watch going around, a Patek Philippe that nobody could fix. So he got the watch and opened it up, and it was a gear that was broken. So he made a gear. And so he got known for being able to fix that, [0:10:00] Patek Philippe.

Gibbons: Do you still have a watch from him?

Blum: I do, I do. It's a very important…

Gibbons: How about your mother? Was she a homemaker or did she work?

Blum: Well, both. She was really a homemaker, but when they got to Venezuela, my father kind of depended on her to make money, to bring in the living. She made little clothing for little babies, and little booties, stuff like that, sold them, and that's how they made money there. While my father spent his time building an airplane. [laughs] Which he never had the nerve to actually fly. [laughs] It was a glider.

Gibbons: That was in Venezuela?

Blum: That was in Venezuela. When they came up, they came to Miami, and then I was in the hospital there, and then went to the Bronx. I really grew up in the Bronx. That's where, in the Bronx, that I stood on the corner and listened to the two women speaking. The Bronx, it was P.S. 86 in Kingsbridge where I went for the first three grades. And I hated the Bronx. It wasn't so bad, but…

Gibbons: Why did you hate it?

Blum: Because of the bullies. It was hard. You know, it's a kid that doesn't speak the language that everybody else speaks. [laughs] So…

Gibbons: Were you bullied?

Blum: Yes, I was bullied. Very much so. And I was afraid of people. I was afraid of people all the time. I was not afraid of machines or stuff. When my brother Simon, who's very good with people but is afraid of machines, when he put in a wire into the electric socket and got sparks going out, [laughs] "Manuel, do something!" I could go over and pull that out. No fear of that.

Gibbons: It was during this time that you had a bad experience in your school, where the teacher told your mother that you wouldn't be able to…

Blum: Oh. I wouldn't… [laughs]

Gibbons: …probably shouldn't expect you to go to college.

Blum: Right.

Gibbons: Tell us about that – what they thought, what the teachers told your mother, what your mother thought, but what…

Blum: Parent-teacher's meeting. Yeah. They'd given me all sorts of tests, and one test involved the colors. By that time, I was speaking Spanish, and "blue" in English is *blau* in German. But I wasn't speaking German, I was speaking Spanish, and "blue" is *azul*. So I just didn't know any of the colors. I was just… And the teacher says, "Look, he's six years old. He still doesn't even know the colors. He's just not going to be able to make it. You want him to…" My mother wanted me to be an engineer. She said, "You want him to go to college? It will be lucky if he can get through high school." It was devastating to my mother. Not so much to me. Actually, I mean I think my parents protected me so I didn't know that "Oh, they…"

In fact, it was something else. My mother told the teacher, "No, he's really smart. He just doesn't speak English." So the teacher said, "Well, speak to him in English," and my mother said, "But I don't speak English." The teacher said, "Well, what do you think you're speaking to me?" [laughs] And we switched, and you can believe it that a month after switching to English, I'd forgotten all my Spanish. I'm good at learning languages. I'm even better at forgetting them.

Gibbons: You also said that you remember becoming interested in brains and that you wanted to be smart about this time. Tell me about that. When do you remember getting interested in brains?

Blum: Yeah. That was because I was having trouble in school, and also words, it's funny spelling words in English. Well, I was having trouble in school and basically I asked my father what to do. He said, "Well, just memorize." So I asked him, "How? How do you memorize?" It really bothered me that we didn't have a manual to explain how to do this. So I got a very good suggestion from my father. He said, "If you understand how the brain works, then you'll be able to be smart." I really liked that idea. That's how I started thinking, "Oh, I really would like to understand the brain."

Gibbons: How did you go about it? Did you read about the brain? Did you have hobbies or anything that helped you go after that?

Blum: Yeah. The sort of stuff that interested me was the kind of stuff that interests you. Ancient man. You know, the Java Man, Neanderthals, the Cro-Magnons, which… I really would read as much as I could about these ancient peoples. I remember in fact at the kitchen ta-… sitting at dinner and standing up and telling… No, not standing up. I was sitting at dinner and I told my parents, "You know, I want to be an anthropologist." And my father stood up – I'd never seen him do this before – and said to me, "You will *not* be an anthropologist. Anthropologists are teachers and teachers can't make a living." So that was that. I was to not be an anthropologist, I was not to be a professor. It's amazing that I became professor.

Gibbons: How old were you at that time do you think, roughly?

Blum: Well, it's between… Yeah, it's fifth grade. Five and six is 11. Eleven years old.

Gibbons: So they wanted you to be an engineer?

Blum: They wanted me to be an engineer. They had me going to be an engineer, my brother Simon would become a lawyer, and my brother George would become a doctor. [laughs]

Gibbons: It's a good plan. [laughs]

Blum: It was a good plan. The best of plans… you know?

Gibbons: Very much an immigrant story, them having this plan.

Blum: Uh-huh. Right.

Gibbons: And so then during your years of school, what excited you intellectually? Did anything spark your interest when you were in junior high or high school? You went to which high school in New York? Where did you go?

Blum: That was military school. But the school that I went to, I went to P.S. 86 first in the Bronx. Then we moved up to Westchester, P.S. 15. I remember that what I loved was science and math. Those were my two loves. History I found very difficult, English. But science and math I could do.

Actually, I remember in the third grade getting a report card. My report cards normally came back "U," "U," "U," "U." Then one, must be in the fourth grade, came back "U," "U," "S," "U," "U," "U." The "S" was in arithmetic and science. I asked my father, "What is this?" I mean I had no idea. He said to me, " 'U's are good, but 'S'es are even better." That gives you some sense of actually what a bad student I really was. [laughs]

Gibbons: But it gave you some sense of what you were good at.

Blum: Yes. I knew that I loved the math and the science. I remember the teacher even asking the class how many atoms do we think there are in the body. The kids were answering, "One," "Three," "Two." And I was so pleased with myself. I said, "A million!" And she answered, "No, no, no. It's millions of millions," which is correct. [laughs]

Gibbons: And why did you go to military school and what was that like?

Blum: After this grade school, sixth grade, we went down to Venezuela for several years, and that's when I learned Spanish. Then my father wanted me to… He said he wanted me to go to a school in the US so I could get ready for college. So I went there.

The military school, it was Peekskill Military Academy. [0:20:00] Lots of people there really loved it. I hated it. I hated it because everything was memorization and I wasn't learning how to do anything. The teacher who taught me algebra, a good teacher, would say, "For this kind of problem, you put an equals sign, two lines, and you put this number here and this number there, and explain what you multiply and divide." And this worked. This worked for the kids. They got good grades from this. But I had no idea what was really going on.

The way I learned what was going on was by going to MIT. I was so lucky to get accepted there. I mean just amazing that I did get accepted there. I don't know what they could have seen, but anyway, I got accepted. And it was such a thrilling learning experience. It was the hardest year of my life, this first year at MIT. Just impossibly hard. So I have many good memories. One of them was a friend of mine coming in and seeing me studying physics, and I was clearly memorizing the physics. He told me, "You don't memorize. You know that F = ma and then you derive everything you need from that. You don't memorize." That was such an eye-opener – "Oh! That's how you do it." I loved that. That was fantastic. And my grades went from being, I remember physics a D grade, and I went up to A's. It was that little thing which somehow nobody had ever mentioned to me. And it was really what I wanted. I mean I loved this idea of being able to actually understand something and derive the stuff yourself.

Gibbons: It's amazing to me given the education you had that you did get excited about intellectual pursuits and being a creative, out-of-the-box thinker. What do you think kept you that way, at least that allowed you to be that to get to MIT to do that?

Blum: [laughs]

Gibbons: Is there any spark in there that you think helped give you confidence enough to do that, to keep on your path?

Blum: Why did MIT accept me? I cannot for the life of me [laughs] answer that question. I have no idea why. And, you know, my first grade in physics, which I just spent huge amounts of time memorizing, trying to do, was a D+. My teacher back in military school, my principal sent me a letter saying, "This is very disappointing. You should be working a lot harder." [laughs] Should be working a lot harder. That was rough. But it was nice that once I got this idea that you derive everything, my grades went from D+ to B and then up to A. I really learned something important there.

Gibbons: What you learned was how to learn…

Blum: Was how to learn, yeah.

Gibbons: What excited you most intellectually? Then we should get to some of your mentors in what you worked on. But what were the sparks that excited you most? And I'm obviously looking for…

Blum: There were lots of wonderful, interesting problems. It's like the Zen koans, but these were mathematical problems about "You have 13 coins. One of them weighs a different amount from the others. You have three weighings. Find the coin that's bad and whether it weighs more or less." These kinds of problems, they were the kinds of thought problems that I… And I really loved them. I remember being given a problem by another student. I learned a lot from the students that were there. Another student gave me a problem of the monkey and the coconuts. You know this problem? [laughs]

Gibbons: Yeah. Tell us the problem.

Blum: It's a lovely problem. The problem is that these five people are shipwrecked on an island with lots of coconuts and a monkey. They decide they're going to wake up in the morning and divide up the… they gather the coconuts, they'll wake up in the morning and divide them up amongst themselves. Well, in the middle of the night, one of them wakes up and decides, "Hmm, I think I'll take my share now." He divides it up into fifths and takes his fifth. Then the next person wakes up… Oh, it's actually a little more. Takes his fifth. There's one left over which he throws to the monkey. And the next person wakes up, sees the pile of coconuts, divides it up into five, takes his fifth with one left over, which he gives to the monkey, puts the coconuts back together again, and so on. This goes on until the end, till the last person. Then in the morning they wake up and they divide the remaining coconuts, and it divides evenly. Each gets exactly a fifth. How many coconuts? The question is "How many coconuts?"

This was a wonderful problem. I spent all of Thanksgiving vacation I remember thinking about this problem. It just drove me nuts. It's just a shame that I didn't know how to think about it, that there are many problems in which you should think about it like this. Start with just two people shipwrecked. Answer the problem if there are two. Once you have that, then try for three. [laughs] This is actually *the* idea that has carried me through since. Any time I get a problem, the first thing I try to do is start with a case of $n = 1$, then $n = 2$.

This was something that I could have been taught in grade school. In this military school, we got probability problems. Nobody could answer them. Nobody. And yet there's some very simple thing I could tell those kids. You get these problems of "You have 30 people in the classroom. What's the probability that two of them have the same birthday?" Well, the way you do it is you start off with three people in the classroom and five days in the year, how many people have the same birthday? You start small and then you get the idea and work your way up. This works for every single probability problem that we had there. I would have loved to be given that insight into how to solve these problems.

Gibbons: When did you get that insight? How did you get that insight?

Blum: Well, it was clearly not right away, because I spent the whole of that Thanksgiving week trying to solve the "monkey and the coconuts" problem, which would have been easy had I started small. Where did I get that insight? Somewhere along the way. It's hard for me to say. I just know that some of the problems that tortured me then, I look back at them and say, "Why was I so tortured? It's so easy to solve them."

Gibbons: So you have talked about some really important mentors you had. You described Warren McCulloch in particular and Walter Pitts, and working in their lab. Tell us the story of how you got there. I think you took a class first on Freud?

Blum: Right. It really began… There are these humanities requirements at MIT. You have to take a humanities course. I was able to take a course with somebody in Western civilization. This person was Professor Schoenwald, Richard Schoenwald. He was great. He was wonderful. He is one of the people that really taught me for example how to look at a picture – "Spend some time looking at it. Sit there, look at it. You will see things come out that are not immediately evident at the start." Just I loved him. I loved him.

Then I was really interested in consciousness and the brain, and his PhD thesis had been on Freud. So I took a reading course with him. Two semesters. We read through the entire 24 volumes of Freud's works. Too fast, actually. But anyway, it was great. I was definitely a Freudian. I thought that this was the way to find out about the brain.

Then maybe a year later, [0:30:00] he stopped me in the hallway, went and said, "You know, there's a person here named Dr. Warren S. McCulloch, who is the anti-Freudian." The anti-Freudian. Where Freud had written the future of an illusion, McCulloch had written the past of a delusion. So "Why don't you introduce yourself," he said, "to McCulloch?" I went down and I did that. You know I read the stuff that McCulloch was writing. It was very interesting. It involved neurons, it involved some interesting problems. I was able to prove a theorem. I went down and told him about this theorem.

Gibbons: What was McCulloch known for? What did he do before you got to his lab?

Blum: Oh yeah. So it's McCulloch and Pitts did this amazing piece of work. First of all, McCulloch defined a formal neuron. It has inputs, some positive, some negative. The neuron adds up the positive and negative inputs, and then compares to threshold. If the inputs add up to a number greater than threshold, it fires. Otherwise, it doesn't fire. This was a simple model of the neuron. Really simple. What they were able to show is that a machine built up out of these neurons could in fact essentially be a computer, could simulate a Turing machine. Such a machine with paper tape could simulate a Turing machine.

First, this was at that time, 1948 when they came out with it, a great insight. For one thing, the neurophysiologists at the time said, “Hey, we’ve seen excitation to neurons. We’ve never seen inhibition.” They wouldn’t accept that, but McCulloch said, “It has to be there. I can show you that without inhibition, it’s just not possible to do the computation. You’re very limited.” So they looked and sure enough they found inhibition in the brain. That was one of the nice things that McCulloch had found.

There was another wonderful thing he said. It shows again what he had to say about Freud. He said, “Given a machine,” what we would call a finite automaton, “Given a machine and knowledge of the inputs, you can determine what the output will be. But given the machine and the output, you cannot determine the input. There are many inputs that could have led to that output.” He used that to say, “The Freudians are trying to go that direction, [laughs] trying to have you recall what went on in your youth,” and he was basically saying you can’t do that.

Gibbons: Were you converted in your thinking?

Blum: Oh, he completely converted me. [laughs] It was wonderful. It was also wonderful because he really wanted to be able to build these machines. This was great. This was what I wanted.

Gibbons: Was he in EECS? Was he in Electric… In which departments did…?

Blum: No. He was in the Research Laboratory of Electronics. I don’t know if it was EECS. He was brought there because of his very exciting ideas. There were many “really exciting idea” people there, and McCulloch was absolutely wonderful that way. And really my mentoring, the mentoring I do is because of McCulloch. McCulloch, he would listen and he would be very supportive. He really would say, “You know, you’re good. You can do it. You’re great. You’re better than everyone else.” [laughs]

Gibbons: He told everybody that?

Blum: Yeah, of course he told everybody that. It was fine. They all were supported by it. They all really were encouraged and did some of their best work because of it. He was great that way.

Gibbons: You have said he was your most significant mentor.

Blum: My most significant mentor, yes.

Gibbons: Why do you think of him that way?

Blum: There are just lots of stories I could tell. I'll tell one. He took me to one of these large conferences and he gave a speech at this large conference that was attended by thousands of people. It was just amazing. Just had people roaring and learning at the same time. One of his stories was "I'm a Connecticut farmer." He did have a farm. "I'm a Connecticut farmer and I'm used to going to the fair where they try to decide which pig should get the blue ribbon. The way they decide this is they find a rock and they put a plank across the rock, and then they put the pig on one side and then they put rocks on the other side until the rocks balance the pig. Then they guess the weight of the stones and calculate the weight of the pig." I'm not saying it the right way. He was trying to make a point about how science is sometimes done, this idea of you guess and then compute. It was just really well done.

Gibbons: So he encouraged you, but he also showed you this mixture of inference and rigor it looks like.

Blum: Yeah.

Gibbons: His style of excitement about his work, that seemed to have an impact on you.

Blum: Yeah. I was taken into this neurophysiology lab, which was great, and the people there were great. I mean there was Warren – he was the centerpiece – but there was Walter Pitts, who was a mathematician. I learned some wonderful things from Walter Pitts. For one thing, that everything you want to know is in books. You want to know something? Just find the right book. Nowadays, just Google for it. Isn't it wonderful that nowadays, any question you have, you can Google for it? Five people have already asked the same question and three people have answered it. It's just so wonderful we can do this. Walter really taught me for example about books, that you should read books. He would go to the library and take out books and books and books until the library got angry and told him to bring the books back, and he would get a wheelbarrow to bring the books back to the library.

He also told me, taught me about how to… There was a problem. Warren had… I called him "Warren." Dr. McCulloch, Warren McCulloch. Warren had posed a problem, the kind of problems I love. He said, "You take a cube and tie a thread to one corner, and then drop the cube through a table. Just drop it through until it's halfway through. Drop it through a plane until it's halfway through the plane. Just stop at that point. What's the figure on the plane? What's the figure that's being produced by this cube?" Wow, that was very nice. You can think about it. You eventually can get what the answer is. But I couldn't see a really good way to prove the answer. Had to use intuition. I mean the answer is it's a hexagon there, but I couldn't see an easy way to prove it and to use my intuition.

And I asked Walter about it, Walter Pitts, and Walter said, "You use algebra. You use the visual for seeing what's going on, but when you want to prove it, you do it algebraically." I said, "Really? Algebraically? Algebraically? This is a very visual problem." "You do it algebraically." And I've since found that other really good mathematicians basically do do it that way. There's Stephen Smale, who proved that you can turn the sphere inside-out. [0:40:00] It's a complex way to do it so that you don't get a ridge when you're turning it inside-out. There's a sphere, you can pass it through itself, you get a ridge if you do it the obvious way. But how do you do it without getting a ridge. Steve Smale proved that you can do it without. I asked him, "How do you do it?" "Algebraically." He never visualized it. It was done algebraically.

Somebody else actually made up these wire-net models, you know the kind of wiring that they use to… chicken coop wires to actually show this, the transformation as it's being done. Later on, Nelson Max came and spent a whole month doing measurements on these wires and then making a movie. It's a wonderful movie. It shows the sphere turning inside out. Sometimes it's transparent. Sometimes it's not. Sometimes it's red on the outside and blue on the inside. Sometimes it's a grid. This is a movie where he shows it again and again and again, and then it stops. I get the feeling when it stops, "If only he would show it one more time." [laughs]

Gibbons: So what did you do in that lab and why was that important to your thinking, in addition to learning to use algebra to solve these problems?

Blum: Well, really there were neurophysiologists at that lab and they actually took me under their wing. One of them was Jerry Lettvin. Jerry had written this paper, "What the Frog's Eye Tells the Frog's Brain," a wonderful paper showing that the eye sees four different kinds of things. One of them is something moving around like that. Then there's a neuron there that goes "Brrr!" when it sees something. If something goes across, it doesn't see it, but that neuron is looking for this. That's the fly that the frog is going for. Then there's another neuron that fires when it sees a cross going across the field. A long stick and across. If the cross is back… if the long part goes first, it doesn't fire – that's a goose. If the long part is in the back, it fires violently – that's a hawk, and the frog when it sees a hawk will jump.

This was wonderful what he had shown. He would show me again and again the kinds of experiments that he had learned from Helmholtz, who really knew about the eye. One of them, very simple, when you're out on the beach, look up at the sky, a blue sky, no clouds. Look at the sky, just stay looking at it, and you will begin to see these little white things moving around. I don't know if you have noticed lots of little white things travelling around in the sky. What is it? Those are the blood corpuscles that you're seeing. The retina has the arteries and veins and nerves. Really they're on top of the retina. It's funny. We see through the arteries and veins, and we only see things that move. We only see things that move, so normally you don't see those arteries and veins, nerves, because they don't move relative to the retina. But the corpuscles, they're moving, and when you look at the blue sky, you'll be able to see them wending their way in front of the retina. It was great. This was one of the many things he told me.

Another one, which I really loved and then did this one myself, he took a battery and connected two light bulbs to the battery with a switch. If you switch it one way, you get one light bulb turning on and the other one turning off, and then you switch the other direction, it reverses. So each light is going on and off complementary to the other. He had made this. He put it on my eye – "Just close your eye" – and he put one light bulb on one side of the eye, one light bulb on the other side of the eye, and then he started switching. What I saw was suddenly that my eyeball had blasted to smithereens and I tore this away from my face. I was sure that I had lost my eye. He pointed to me and said to Warren, "You see, he saw it." [laughs]

What had I seen? Well, apparently these veins and arteries, we don't see them because they're not moving. When the light is going on and off, the light will cause a shadow on one side, then a shadow on the other. Back and forth, shadow. Shadow is moving, and what you see is that shadow. It's a wonderful demonstration.

They had lots of stuff like that. I was learning how these biologists think.

Gibbons: Were you still in electrical… You were in electrical engineering at that point?

Blum: Right. And that was good in this neurophysiology lab, because Jerry Lettvin was probing these frogs' eyes with electrodes and Pat Wall was there and he was probing cats' brains with these electrodes. I felt pretty upset about the cats, but nevertheless he was probing them. Pat Wall was very important too. He has some very good suggestions about pain and how we get to feel pain and how to stop pain.

Gibbons: And that's of interest to you now.

Blum: And that's very much of interest.

Gibbons: We should come back to that later when we talk about consciousness. But it's interesting that in that lab, you were learning about the brain and how it processes inputs. Did you work at all on these very early neural networks at that point too?

Blum: Yeah, that's right. McCulloch was interested in the problem of how the brain is able to do its work without error, with essentially no error. Right? We don't suddenly reboot. We don't fall down in the street. We do pretty well. He was puzzled by the fact that when we drink coffee, the threshold of our neurons all go down, so more of them are firing, and when we drink alcohol, they go up, so less of them are firing, and nevertheless we are still able to walk and we're still able to talk. He was really puzzled by that. That actually came from von Neumann. Von Neumann wanted to understand that problem and Warren said that von Neumann would kick him in the shins to come up with a solution. Von Neumann had a solution, but it required millions of neurons to simulate one good neuron. Millions really of bad neurons could simulate one. This obviously couldn't be the answer. And the nice thing is that in Warren's lab, we got a really good idea how it's possible with just a few neurons. Just a few neurons can now be put together in such a way as to simulate a perfect neuron.

This was the sort of stuff I did with him. It was a time when Claude Shannon had done information theory, and information theory was very important for debugging communications, for making sure that… You know when you send signals down a line, there will be bugs and errors. Information theory is what makes it possible to nevertheless be able to get an error-free communication.

Gibbons: From there, that lab, very important to you, then went on to… Was this when you went and studied with Hartley Rogers and began to think about computability theory?

Blum: Yes. That's the mathematics.

Gibbons: Yes. Tell us about that.

Blum: Wait. Before that. [laughs] Before that, I want to mention that McCulloch was fantastic, but there were lots of fantastic people there.

Edwin Land was there. In one of his lectures, he showed a picture of a bowl of fruit with lots of different colors. He shows this picture through a red filter and then through a pink filter. Two different filters, [0:50:00] just slightly different, superimposed, and when they superimpose, you saw all the colors. All that was, was red and essentially pink, which is red and white, and then when they're put together, you see all the colors. We're supposed to need three… We're supposed to be able… We need to see red, blue… cyan, magenta to be able to see all these three colors. What is it? Magenta, cyan, and yellow. You need three colors to be able to see all three. Land was showing you no, you don't need all three. You can get by with just two different ones. How? How is that possible? Very exciting. Actually, if you look in Wikipedia at Land's work, you find out a little bit about how this could be. It was just very interesting.

There was another person there that was wonderful, Manuel Cerrillo, a Mexican engineer. I'd go to his lab and he'd show me the latest thing he was doing, which was always so amazing. He had built himself a hi-fi set, a wonderful hi-fi set. He took a record and put it on the turntable, and I could hear the band playing and the woman singing. Then he turned a knob and the band disappeared. You heard only the woman singing. Of course, you turn the knob the other way, the woman disappears and you hear only the band. He was doing this at a time… This is before transistors. The circuitry he had to put together by hand – resistors, capacitors, and vacuum tubes – to make this thing. It was just an amazing thing that he had done.

Gibbons: Did you build things too? Were you interested in…?

Blum: I was very interested in building things, but I didn't know how to do it. The unfortunate thing about an MIT electrical engineering education is that you don't learn how to build stuff. In the lab, when something had to be built, they would hire an engineer from Northeastern. A student from Northeastern could build the circuit that they needed. The MIT student couldn't do it. They didn't know anything about building. [laughs]

Gibbons: Hired out. [laughs]

Blum: Hired it out.

Gibbons: Oh, it's amazing. So you were in electrical engineering but working in these very interesting labs.

Blum: Working in these very interesting labs.

Gibbons: … learning about the brain. Were there any other things you were doing before you started getting into Hartley Rogers' lab? Because I know that's important to talk about too.

Blum: Well, I wanted to understand the brain. Especially I wanted to understand consciousness. I wanted to understand, you know, consciousness. Let's talk about that. I wasn't allowed to think about that. Wasn't allowed. Walter Pitts told me, "Look, the only way we can information about what's going on in the brain right now, except in the rare surgeries that are done, is with EEGs, and the EEG is looking at voltages through a skull that's that thick. What it's measuring is just some average of voltages that it's seeing. You're just not able to find out what's really going on in the brain." So talking about consciousness, thinking about that sort of stuff was simply not permitted. Which was really a shame because I really wanted to understand that.

The great thing is that since then, we have gotten much better ways to find out about how the brain works. Right? There's now this fMRI, functional magnetic resonance imaging, that first was invented in 1989-1990. For me, not that long ago. Of course, you were born after that, but for me, that was relatively recent. That allowed us to see what was going on in the brain.

Gibbons: Do you remember consciously being frustrated that you couldn't work on it at that time and thinking, "Well, I have to work on something else"? Or did something else just naturally excite you and you moved in that direction unconsciously?

Blum: [laughs] No, it was just something I would come back to every so often. I had a wonderful student, Ryan Williams. I talked to him about consciousness. He got these huge bulletin boards of information that he put up in the halls where a history of thought on consciousness in the brain. We talked about it and he was excited by it, but we never got anywhere with it. His thesis was a great thesis, but it wasn't on that because we didn't know how to pursue this.

Gibbons: Does this say something about how timing is important…

Blum: Huh. Good point.

Gibbons: …that a good idea's important but the timing also is when you're trying to tackle it?

Blum: Yeah, for sure. This whole question of… An example of the kind of question we would come up with is free will. Free will. You know, we're free to choose what's best for us. This is a problem that's come up over the ages. A century before Christ, you see a philosopher-poet Lucretius writing in Latin of course *De rerum natura*, *On the Nature of Things*, where he writes beautifully exactly the question "How is it possible in a world" – and this is before Newton – "in a world where atoms don't move unless they're pushed, where the action of everything is determined, how is it possible in this determined world to have free will?" He stated it much better than I've stated it.

Then later, around the time of… a little after Newton, contemporary with Newton, there was this English Dr. Samuel Johnson, who basically said exactly the same thing. He said, "All science is against the freedom of the will; all experience is for it." He's clearly pointing to this very deep paradox – how is it possible that science is telling us that everything is determined and our experience is we have free will, we can choose?

This sort of question really bothered me. I really wanted to know the answers. The wonderful thing is I'm beginning to see what the answer to that question is.

Gibbons: We have to come back to this. This is really important. Let's take a break for a minute.

[Recorder is paused briefly]

Gibbons: So Manuel, how did you get interested in mathematics?

Blum: I was an electrical engineer, but I hadn't learned how to build the circuits, and in any case, I was kind of being steered towards mathematics by Walter Pitts, a mathematician who was really giving me a handle on how I might do something mathematics. So I went into mathematics, and that was very hard for me. It was really hard. But eventually, I learned how to prove theorems. It was amazing. I was sure that proofs aren't real. I mean you get this intuition about something, you get an understanding of why something is true, and then you have some words. "Are proofs real?" I mean… And yes, I was told, "These proofs are real. There really is a logical way to decide whether or not something is true." And we kind of know it now because there are these proof checkers. If it's a really good proof, then this proof checker will actually be able to say, "Yes, it's a good proof," and if it's not good, it will say no. This was something that I just hadn't understood.

So I got into mathematics. These teachers were wonderful. They managed to convince me that this was a real thing that you could do, really prove theorems. Then in order to understand the brain, well, the brain is these neurons working logically together. Logical circuits. Just somehow seemed to me that logic was the right thing to get into, and there was a part of the logic which was concerned with computability. It was called recursive function theory, but it really should have been called computability theory, theory of computability. And there was a professor, Hartley Rogers, teaching it, and he was a very good teacher. So I [1:00:00] tried to do that.

And it was hard. It was also hard. I remember turning in these homework problems and then getting back "Wrong." The wonderful thing was the TA who actually graded these problems had a thought about the proper way to grade. He would not consider a proof wrong unless he could give a counterexample. You know when you prove a theorem and somebody says, "Well, this is not right," but you're pretty convinced it is, that's not convincing to say, "Wrong." But if he gives you a counterexample, "See, here's something wrong with your proof. It doesn't take care of this example"… And he only would take off credit if he could produce a counterexample. This was wonderful. Wonderful! Absolutely amazing.

So I began to learn recursive function theory and it was very exciting.

Gibbons: Can you explain what recursive function theory is?

Blum: It's about what Turing machines can compute. What they can compute, what they cannot compute. These recursive functions, really computable functions, it's what functions can be computed, what functions cannot be computed. It's a beautiful theory for trying to separate out those two. So I went to Hartley Rogers' course and then I took it, and then the next year when he gave the course, I went to it again. I went and did it again. It was just wonderful.

Gibbons: What were the problems you worked on then? What did you get most interested in?

Blum: I was really working in McCulloch's lab at the time, but these problems that came up were interesting. Then I had to pick a thesis advisor. There was a person there that was just ideal, Marvin Minsky. He really… He's one of the two… Minsky and McCarthy coined the word "artificial intelligence," and he was really interested in how brains work. It was great. So I attached myself to Marvin and he became my thesis advisor.

Gibbons: How did you meet Marvin and how was he perfect? Not just being interested in the brain, but how was he working on that problem that intrigued you?

Blum: Well, first, one of the things is that Marvin would come down to see Warren McCulloch every so often. I remember one particular time, you know in logic you have this notion of a Venn diagram. A Venn diagram on one variable is just a circle. If you have two variables, it's two overlapping circles. You see all possibilities – A alone, B alone, A and B, none of them. Then you overlap three circles, you can get all combinations. Four we know how to do, but we didn't know how to do five or six or seven.

So Marvin came down and showed us how to build a Venn diagram with any number. *Any* number. It's an absolutely wonderful idea. I don't know how to say it, to show it. I could show it on paper.

Gibbons: How important is it to what you did after this?

Blum: No, it's important only because I was so… it's an example of some of Marvin's ingenuity.

Gibbons: Yes, yes.

Blum: But we can go…

Gibbons: That's so interesting. Then so because you worked with Marvin, it changed… you started working on more concrete complexity. Can you tell us what this is and why you switched, what the change was intellectually at that time?

Blum: Yeah. When Rogers taught us that it's possible to build a universal Turing machine, this was with words and explanations of how this thing would work.

When Marvin did it, it was by actually designing an actual logical circuit for a universal Turing machine. It had 25 states. It had these pieces to it that were completely designed. He has this book *Finite and Infinite Automata*, and when you open it, there is the design of his universal Turing machine.

He was very concrete, for one thing, and there were some things that were really important to me that I learned from him. For example, von Neumann had proved that it's possible to build a self-reproducing machine. This was going around. This was something that interested a lot of people. Claude Shannon came in and showed us a very simple self-reproducing machine. It consisted of… It was a wooden thing. Very simple. It was maybe how… It doesn't matter. The point is the ideas were around and von Neumann had given a proof that it's possible to build a self-reproducing machine, and I couldn't understand it. It was a hard proof and I wanted to understand it.

Marvin explains it through a homework problem. Absolutely amazing. Just wonderful. He gives as a homework problem to build a self-reproducing machine, and he gives two hints. Those hints really lay out how to go about it. It's actually beautiful. I'll tell you. You first build a machine that can look at any blueprint and construct a machine following that blueprint. So it's a universal constructing machine. It can look at a blueprint and can construct whatever is there. Then what you do is you have the machine look at itself to get the blueprint, put it down, and now copy that, construct that machine. Two ideas, the universal constructor and then looking at yourself. It was just terrific. It was the right way to teach this sort of stuff.

Gibbons: What was Minsky like to work with as a student for you?

Blum: He had a laboratory full of machines. He made the playthings available. It was very good that way. In fact, when I would come into his office, invariably I would find him working on a hand. He was interested in hands, and he'd have all kinds of hands. That was his big thing.

When Stanley Kubrick wanted to do *2001*, he came by to talk to Marvin about hands. He was going to have in his film a robot, and how should the hands work? Marvin said… And he would have a different answer every day, but this particular day he said he would build it like this. He'd have two arms that could do this to pick up something. Then at the end of the two arms, there would be these pincers that could do this to pick up smaller stuff. And at the ends of those fingers, he'd have more pincers, smaller pincers to pick up really small stuff. And he said he'd do it this way because the program that works for the two arms then is the same program you can use for the hands [laughs] and the pincers. So that was nice, and you see that in *2001*. There are these pods that go out there and you'll see their arms working that way.

Gibbons: What was your thesis? This is still in electrical engineering, but what was your thesis?

Blum: No, no, no. Now I'm in mathematics.

Gibbons: You're in mathematics as an undergrad?

Blum: No, no. I was…

Gibbons: This is, we're now to your PhD?

Blum: This is now to the PhD. So the PhD…

Gibbons: Okay. We're in the PhD years. This would be… What years are we talking about now?

Blum: The PhD, so 1961 to '62-63.

Gibbons: Let's talk about that. How did you get your idea for your PhD thesis?

Blum: I was in McCulloch's lab. It was good. And yeah, so how… I was really trying to use this recursive function theory to be able to understand brains. One of the things that happened is there was a professor, Michael Rabin, that came and gave a course at MIT. He was just wonderful, a wonderful thinker, and he had proved a theorem. The theorem was that there are functions that are hard to compute that simply cannot [1:10:00] be computed faster than a certain amount. No matter what function you give it, what computable function you give it, you can come up with a new function that can't be computed in that amount of time.

So this was a theorem. He gave me the paper. The paper was missing the proof. [laughs] He was surprised and upset that he'd given me… when I later showed him he had left out the proof. But the great thing for me is that I could spend several days thinking about it, and eventually came up with a proof, really with the proof. Again, that's an example where you know you get somewhere by not being told the answer.

Gibbons: By giving important clues.

Blum: By giving clues, in this case a statement of the theorem. Once you have that theorem stated, then it was interest-… the proof enabled me to think about something else, which was a question of how efficiently can these functions between computed? I wanted to have some notion of optimality. A function can be computed… I'm thinking about a function like *f(n)* and it's an integer, *f(n)* is the n^{th} prime. This would be a computable function. And I wanted to be able to have some notion of how to define "optimal program." Sometimes some programs are not optimal because they're faster programs, programs that can do the same thing but faster. And I wanted to have some notion of when is a program really about roughly the best you can get, when is it optimal?

One thing that came out of this study with Michael Rabin… Michael wasn't there, but one thing I realized from it was that in fact something very counterintuitive, that for example, I'll give you an example, you might say that a function is optimal if there's no program to compute… a program is optimal if there's no other program that can compute it in just a log of the amount of time that this one takes. If a program takes 2^n steps, the log of that would be n. You might say it's optimal if there's no program that will compute it that much faster, that will just take a log.

What I was able to show is no, that there are functions with the property that, whatever program you have for computing it, there's another program that is much faster and takes just the log of the number of steps. That one's faster and then there's another one that's even faster that takes the log of that. I remember talking to Shmuel Winograd, who was really, really good, and I said, "What do you think? Do you think…" [laughs] He was really amazed also that there should be, that there should exist these functions that simply cannot be computed in an optimal way when you define it in this case using logs.

So that was the Speedup Theorem.

Gibbons: This was the heart of your thesis, the Speedup Theorem?

Blum: Yeah, it was an important part of it.

Gibbons: How was it received?

Blum: No, it was received pretty well. For one thing, once I had proved this theorem, I could see that the proof relied on some very simple stuff, which I took as axioms and then just proved the hell out of it, used those axioms to prove as much as I could prove. It gave a very nice theory. But this theory was very abstract and not very useful. I mean it's kind of nice to know that it's hard to define optimal, that there's a serious thing here going on, they can't define optimal. But it didn't do what Steve Cook's P-NP does, which was really using recursive function theory, the same recursive function theory to get a real handle on the kinds of problems that people really are interested in computing.

Gibbons: Did that bother you at the time or did you pursue this…?

Blum: Well, I was just going in my direction because everyone was encouraging me to do that. But in fact the Cook theory was really the right one. Did it bother me? It bothered me that here he was making use of exactly the mathematics I knew, and why didn't I come up with it? But wow, he did such a good job, Steve

Cook. Beautiful. Then Dick Karp also showing how this applied to problem after problem that came up in the real world. You know, the Travelling Salesman Problem, graph-coloring problems. Many different problems all turned out to be NP-complete, the hardest amongst these problems in NP. So here was the logic being used to actually lay a foundation, and then somebody, Dick Karp, who really knew what sorts of things people are trying to compute then showing how it applies, directly put the two together.

Gibbons: Many people do abstract computability. They're doing this kind of theory and they're happy to do the abstract thinking. Did you feel a pull or a tension that you wanted to do things that were a little more practical? Was that important to you?

Blum: Well, I'm an electrical engineer. [laughs] I'm an electrical engineer more than I am a mathematician. I love this practical direction. Yes, it was really very important for me. And in fact that's what I want to do with consciousness. I mean I want to really lay out the blueprint for building a conscious machine. I'm not a neuroscientist, even if I did work in a neurophysiology lab. I don't go into laboratories and actually do experiments. I'm very happy that other people do. So what could I possibly do to contribute to understanding consciousness? Well, maybe I could use my knowledge of logic and mathematics to lay a foundation for building a machine that is conscious.

Gibbons: Let's go into that. Let's go ahead and talk now about what you want to do with consciousness, then we'll come back to your time after you wrote your thesis and went to Berkeley. Do you want to do that now?

Blum: Yeah, yeah. Sure, let's do that.

Gibbons: Because we're going right into it, it seems like we should do that.

Blum: [laughs] Okay, sure.

Gibbons: Tell me what you're working on now and when you started working again on consciousness, your theory of consciousness.

Blum: Sure. When I had first gone into McCulloch's lab, I had wanted to understand consciousness, but I was told by Walter that there's no way that one should look at that. In fact, there are plenty of people now that think that that's out of bounds and we shouldn't be thinking about consciousness. But there are some really good neuroscientists that have come up with something that they call the neural correlates of consciousness, stuff in the brain that corresponds to at least some aspects of consciousness. There's a part of the brain called the amygdala. This is a really interesting part of the brain. It deals among other things with fear. If you're in a fearful situation, the amygdala fires and says, "Watch out. Something to fear here."

There's a woman whose amygdala calcified. This means that basically she lost the amygdala. It's a genetic thing. So she lost the amygdala. She's afraid of nothing. She is afraid of absolutely nothing, and you might think, "Well, this is very dangerous." I do think it's dangerous. She nevertheless was able to marry, have kids. She knows to watch both ways when she crosses a street, more or less the way we learn to do it – because your mother tells you, "Look both ways before you cross a street" – not because she's afraid. She's just not afraid. People find it very curious to talk to her because she's so open. She loves everyone. [laughs] [1:20:00] A wonderful person. No fear.

So these neuroscientists have found many different parts of the brain that are concerned with many different aspects of consciousness.

Gibbons: And so you are interested… What is your approach to exploring this idea? What are you learning from them that then leads you to want to build "a Turing machine of consciousness" as you described it?

Blum: I should mention that I wasn't working on consciousness and that I'm really indebted to Lenore for… Lenore is really good. She has her ears to the ground, she knows what's going on, and she told me, "It's time."

Gibbons: This is your wife, Lenore Blum, who is an eminent mathematician in her own right. You had always wanted to work on this…

Blum: And she said, "You've always wanted to work on it. You've always thought about it. Now is the time."

Gibbons: And this has been – what? – your great disappointment sometimes in life, that you didn't get to work on it? Or…?

Blum: Well, it was disappointing that I was born into a time when you weren't allowed to work on it, when it was simply off-limits. But she said, "It's within limits now. Not everybody believes it, but there's good stuff being written. Time to look at it." And in fact that's the case. It is a perfectly legitimate time, because with fMRI we now can look into what the brain is doing, and these neuroscientists are finding these neural correlates of consciousness.

And maybe the most important part for me is that there is a model of consciousness that I had been looking for. I know how hard it is to come up with this model because I was looking for it. I mentioned Ryan Williams, my student. We were looking for it. We did not have it. I know it's hard to find it. Bernard Baars came up with a wonderful model for us.

He calls it the Theater Model, Theater Model of consciousness, where he views consciousness in terms of a theater. There is a stage. What we are conscious of is just what's on the stage. There is a stage, there's actors. A few, just a few actors on the stage talking to each other, doing stuff, thinking. And there's a huge audience in the dark, a huge audience of processors. These are the unconscious processors looking at what's on the stage, ready to send information to the stage.

On the stage, the actor is maybe at a party, sees somebody he knows, forgets "What's the name?" finds it… Have you been in that position? You don't remember the name. So, okay, you want the name but it doesn't come, so you go off and you have a drink or talk to somebody else. Then half an hour later, the name pops up. How come it pops up? Because in this audience, these unconscious processors have seen that this is a problem. You want the name of this person. Some processor says, "Where did I meet this person? Where did I see him the first time? Oh, he had cameras. He was at the door. I saw him there." And another processor says, "His name begins with 'B.' " Then finally a processor puts this together and says, "Oh, it's Brian Parker," and that name gets pushed up to the stage and I learn the name of that person. All this thinking is going on. We are totally unaware of it. It is in the background, but it's going on. And if you wonder where these ideas come from, it's coming up from those unconscious processors working their way up.

The model is wonderful. There are these actors on the stage. The entire focus, all the lights are on the actors on the stage. Basically what they are thinking, what they are wanting to know, what they are feeling is broadcast to the entire audience of processors. Broadcast. Every single processor knows about what's going on on the stage. This is very fast and it's being broadcast. Then the answers when they come may come from several different processors. They work their way up to the stage. There may be different answers, there is a competition as to what will actually get up to the stage. It's a much slower process. Only half an hour later do you get the name of that person. But there's this fast broadcast to all the processors of the brain. It's no wonder that what's on the stage is conscious. Every single part of our brain is aware of what's up on that stage, what's going on there.

Gibbons: You talked about how the stuff on the stage is short-term memory and the unconscious is more like long-term memory in this model as well.

Blum: Right.

Gibbons: And this model is rich enough that you then can do what with it? How does this help you design a machine?

Blum: Oh yeah. Good. I really would like to be able to build a machine. There are a number of questions… I guess before that, there are a number of questions that I've had that this model helps to answer, like the free will question. Back to Samuel Johnson's "All science is against the freedom of the will; all experience is for it." So there's an actor up on the stage, as Baars points out, that represents you. That's your self-awareness. It's that actor that's representing you. And that actor is very much like what used to be called the homunculus. It's that actor that represents you.

That actor let's say has to make some decision. Perhaps you're playing a game of chess, and so that actor is being presented with a chessboard, and it's his move and he has to decide what move to make. Should he make this move or should he make that move? Really, you are free to do computation to try to see the value of this move versus the value of that move. You have a certain limited amount of time, and in that amount of time, you may find out which move is better. And at that time when your time is up, you make that move. It's not clearly the optimal, but it's one that you at least decided is most likely the best move. Basically you have free will until that time when you make your move, when you finally decide what you're going to do. That's free will. That's just for me a wonderful explanation of what free will is about.

There are other things. There are many things that one can do to notice that, for example, when you decide which move to make, the decision's already been made. Those unconscious processors have done their computing in the background, have eventually forwarded the decision to you, to the stage. They've already made the decision which move to make before it gets to you. And it's wonderful that we're finding that in fact these decisions are made before we are aware that we have made the decision. There are many things like that which make it clear that this model seems to be correct.

Another puzzling part of the problem for me has always been – I'll mention it in terms of pain – simulation versus actual experience. I say pain because… I mean Lenore would like me to talk about joy first of all, and one could, but pain is something that's very fundamental and it's easy to experiment. I would take a cold Coke out of the fridge and I'd hold it in my hand for as long as I could, and it's very painful. You hold it and you try to understand what's causing that pain, and you get nothing. You cannot figure out what's causing that pain, but you can at least do the experiment. The point is that on the stage, you don't get to see what the processors, the unconscious processors are doing. You can only ask the questions and maybe get the answers.

One of the questions I have is how in fact do we build a machine that will feel the agony of pain, the real agony? And I don't want to just simulate it. You can look, you can Google for "robots that feel pain," [1:30:02] and you will find that robots have been built that really do a good job of simulating pain, but they don't really feel it. I wanted to know how do you *really* feel it? I've been asked when I talk about this, "Well, why would you want to build a machine that feels pain, that is conscious, that feels pain?" I point out that there are some people that are born who don't feel pain. It's called asymbolia. It's got a name. People are born with it and some people get it when they're knocked on the head. Asymbolia is really interesting. If you have asymbolia you can feel pain. You know that it's where the pain is. It could be something hot on your hand. You can feel where it is, how hot it is, the full intensity. You know about it, but it's okay, doesn't bother you. It doesn't bother you.

This is what we can do. We can build machines that are like that, that will know about it, know its intensity, but it's not going to bother them. How does the body, how does the brain manage to get you to *feel* this?

Gibbons: They are unconscious of the pain, so you want to be able to make a machine that is actually conscious of the pain?

Blum: Conscious of the pain. Really, really feeling the agony.

Gibbons: And what will that show you do you think by building it? How by building that will you get a better handle on the consciousness?

Blum: Oh, yeah, yeah. Before I get to that, let me just mention that there are kids that are born with asymbolia, and they rarely live past the age of three. It's a bad thing to have, asymbolia, because if you don't *really* feel it, even though you know about it, you die because you basically… Typical thing is they eat their tongue and their lips and they destroy themselves. They can break a leg and keep on running. They just destroy themselves. The people who get it because they're hit on the head, they know how important it is to pay attention, so they can survive that way. But it's not because they feel it.

So I wanted to be able to build a machine that will feel it. Because machines are expensive. They should be able to take of themselves. And besides that, if I can get the machine to really feel the pain of hurt, I can make it also so it feels pain when it hurts a human being. I can do something that we don't necessarily have. We have mirror neurons which will tell us if somebody's hurting, but I could make it so that hopefully that the robot *really feels* the pain when it's hurting somebody.

Gibbons: That almost suggests a solution to some of the concerns about AI, that if it can feel the pain, will it then be conscious of not doing things that hurt humans?

Blum: Sure.

Gibbons: I mean I don't know that we need to go down that track, but it's a very interesting thought.

Blum: Yes, that's what I would like. So I definitely want to be able to build a machine that feels the pain. I keep track of my ideas of how to do it, and I think I have some handle on it. I'm not saying I have the whole thing. But part of the answer is that some processor in the unconscious, when it knows that there's something that's painful, which is concerned with pain – the amygdala is concerned with fear, there's a processor that's concerned with pain – when it notices this pain, it gets to put that pain up on the stage. Normally we decide what we want to think about. We want to think about a person's name and it will come. But there are processors there for pain, for fear, that have direct access to the stage and they can put up their concerns up on the stage. A little bit of pain, it'll show up on that actor, "There's pain here." A lot of pain? A lot of pain will show up in that you can't do anything else. You simply can't. It doesn't allow you to think about anything else. I think that pain is coming from the fact that these… or at least in part because this processor gets to be up on the stage and doesn't let you do anything else except think about that serious pain.

Gibbons: Where are you in the process of building this model, or…?

Blum: It's good. So first of all, nowhere near building it. Just trying to understand how… trying to put detail onto Bernie Baars' model. I mentioned this theater analogy. As an electrical engineer, as a computer scientist, we would like to build a completely formal model where the words are turned into mathematics. This formal model would enable us to prove theorems. I want to get to the point where I can *really* understand in-depth how the brain is managing for example to get free will and pain and joy and all of these emotions.

That's my goal, and fortunately there are other people that I can talk to about it. Jerry Feldman at Berkeley, I'm talking to him. There's Mark Wegman, who's at IBM. He's a chief scientist in some part of IBM. He's very interested in it too. So we're talking about getting that detail down to the point where at some point we'll be able to actually build this machine.

I need also… See, I need to also understand this mathematically, because without… You know, the way you know that I'm conscious is that you know that I'm built more or less the way you are and you're conscious, so therefore you know that I'm conscious. But how about a dog? Is a dog conscious? How about a cow? How about a worm? Is a worm conscious? Questions like this, we have to have some way of being able to answer them if we're going to be able to answer whether or not the machine is conscious. I need to have the mathematical definitions and the theorems to basically give us some insight into how to tell if a machine is or is not conscious. I don't have the answer to that. I would love to have a Turing-like test. You know, the Turing test is for intelligence. I want one for consciousness, and I absolutely have no idea how to test for it or how to tell if a machine is conscious. I'm hoping that that will come out of this theory that we'll build, some understanding of… a test or some way of knowing from the design why that machine is actually feeling as we feel.

Gibbons: So you're working on this on your own but also with talking with people at Berkeley, where you're on sabbatical this year, and other places? You don't have a student at this…

Blum: Right. Berkeley with Jerry Feldman. Mark Wegman is at IBM. I'm just talking to them. I called up Bernard Baars because I love his model and told him what I'm trying to do, and he's not so sure that I'll be able to achieve it. Which is wonderful. Which is *wonderful*. He's got the theory that tells me you can do it and he doesn't… [laughs] he's not convinced that I'll be able to do it. This is exactly right.

Gibbons: Is that encouraging to you?

Blum: That's very encouraging. He's very nice. He's actually very nice. Very supportive.

Gibbons: Is there anything more you want to say about this model of consciousness or what you're working on or how you want to go with that?

Blum: Let's see. I talked about free will and the desire to understand pain. Of course there's many other things. Part of what I'm doing is trying to understand what it is that you don't need in order to be conscious. For example, I mentioned this woman whose amygdala is calcified. She doesn't feel any fear, but she's conscious. You don't have to feel fear to be conscious.

There's another example, wonderful example. A person named H.M. You heard of H.M.? We now know his name is Henry Molaison, but we called him "H.M." while he was alive. He died only a few years ago. Very well studied. He had an operation in which a large fraction of his hippocampus [1:40:00] was cut off from the cortex. It was removed, actually. And after that, he was conscious, but he couldn't make any permanent memories. He could not make any… Well, he couldn't make any permanent memories. The woman who studied him would come in, introduce herself, they'd work a bit, and then she'd leave. The next day she'd come back, she'd have to introduce herself again. He could not remember who she was.

So I know from this particular person all sorts of things. You don't have to be able to create memories in order to be conscious. On the other hand, there is a memory that he did have. This is this kind of memory where… a memory of a person, a biographical memory, a memory of having met a person, he could not create that. But the kind of memory that you have, the procedural memory which you use when you learn how to ride a bicycle, that kind of memory he could do. So it's very interesting. She would come in. She would say, "Here's a typewriter. I want you to copy this page onto the typewriter." "Okay. But you know I don't know how to type," he would say. Then he'd start typing and say, "Oh, it looks like I can type." He'd been taught how to type and he was a very good typist. He just didn't know that he had learned the skill. Maybe people that are conscious really do need procedural memory. I don't know. But I know they don't need the other kind of memory, the one which you use to learn how to spell words, for example.

So some of the work I've been doing has been trying to figure out what things you don't need to have, and many of the emotions you do not need to have. There is a disease called alexithymia where you don't feel love. If you love… There's a guy that is conscious but he admits that he has never felt love. He married, he has children, but he'll be the first to admit that he doesn't feel love. And when you use fMRI to look into his brain, you see that that part of our brain which lights when we feel this joy of love, it doesn't light up in his brain. It's really great to see that. So he's conscious. You don't need to have that.

There are some things I think you do need to have. One of them is some kind of inner speech or inner vision. Every human being talks to themselves. I've spoken to deaf people. They do talk to themselves. They use their hands, but they have that kind of inner speech. And I'm not saying that you have to have it as English or German or hands. I think that dogs probably have it too. They can imagine, they can see, they can plan to do things. This kind of planning needs some kind of imagery or speech of some kind. I personally believe we can build these machines and I personally believe that dogs really are conscious, every bit as conscious as we are. That's of course just a personal belief. I would love to be able to show from the mathematics why, or that they are, or not.

Gibbons: And whether a worm is conscious or not as well?

Blum: And whether a worm is conscious or not.

Gibbons: Yeah, or an octopus, which has…

Blum: Or an octopus, which has eight brains, one in each tentacle, and then there's a central controller. That's a very different kind of a brain, but I would like to be able to tell if it's conscious. These octopuses are very smart. They can see a glass jar with a crab in it and a top on it, and they've seen you open the top to pull out the crab or to put it in, and now the octopus sees the glass jar with the top on it and it goes over and unscrews it to get the crab. They're very smart. I think it has to plan that. It has some kind of inner speech to plan what it's going to do.

So I'm really trying to get a hold of what are these things that absolutely must be built into that machine? One of them is inner speech. One of them is self-awareness, self-awareness being the actor on the stage that represents yourself.

Inner speech, self-awareness, and the third one is what I call motivation. It's really this energy, energy and desire to do. It could be anything. It could be energy and desire to be famous or energy and desire to make a lot of money or energy and desire to eat. There are many things that this could be, but you need to have that energy. If you don't have that energy, I don't believe you can be conscious. I think to extent that you lose that energy and motivation, you will start to lose consciousness.

Now we'll see. Everything I say is open for grabs until this theory is firm. Even then, once a theory is firm, I'll just be able to say, "I have a model for consciousness and here's what's true of the model." I will myself feel that it's a good model, but it will be up to people to decide yea or nay, to see if the model really does explain what's going on.

Gibbons: It sounds exciting. Are you excited by this and drawn by it?

Blum: Yes, very excited. Very, very much so. And it's great that I have people that I can talk to about it, even though there are plenty of people who will sort of cross their eyes when I mention consciousness. You can just see they're saying, "Oh, this is not kosher." But I've had a lot of good luck talking to people about it.

Gibbons: This might be the time to quote one of your former students, Adleman, who is a Turing prize winner himself, who said, "Manuel Blum, working outside the box, he was a master at that."

Blum: That's very nice of him.

Gibbons: This is definitely outside the box but where you've also made your mark.

Blum: Yes.

Gibbons: You're not scared to take on these very difficult problems and maybe look at them from a side angle or…

Blum: Yeah. Fortunately, I have Lenore to support me. She's very encouraging. She says it's time and she's been supportive of it, and it makes a big difference

Gibbons: She's been a key collaborator. Or maybe "collaborator," maybe that's not the right word because it connotes a work relationship. But tell me about Lenore and her role in your thinking and career. You were married after you finished at MIT? Just tell me a little bit around that.

Blum: Oh right.

Gibbons: How you met Lenore and the part she's played in your life.

Blum: [laughs] I met her when she was nine years old. I was 14. She came in the front door with her mother and I remember thinking to myself, "This is the prettiest girl I have ever seen."

Gibbons: That was in Venezuela when you were back?

Blum: That was in Venezuela.

Gibbons: She's Venezuelan also by birth?

Blum: No, she was born in New York City. She had come down to Venezuela. They were coming by to see my mother. They had just arrived. And the prettiest girl I had ever seen came in the front door there. And Lenore has wondered about this – she was nine years old at the time – "How can you say that about a nine-year-old?" [laughs] The great thing was when our grandson was nine years old, she finally understood, because to her, he was the most beautiful thing she had ever seen.

So we work in different kinds. She works in real mathematics, where "real" is referring to real numbers, whereas the computer science I do is more discrete, more like integers. But she has her ears to the ground and she's very helpful that way. She can listen to my talks and say, "This is not going to work." That's really good. Especially when you're doing something crazy, it helps a lot to know that I'm not making my point very well, and she'll help me with that.

Gibbons: So there's been a lot of support back and forth…

Blum: A lot of support.

Gibbons: …and realistic knowledge?

Blum: Yeah. She's actually doing a lot more than that. I mean she really knows this work that I'm doing. She really has listened and she really has read as well. She's very well read, so she knows what's going on. And that's very important, very important to me.

Gibbons: Yes. How you pick ideas and what you're going to work on and having good taste in what ideas to work on is a key part of…

Blum: Of being successful. [1:50:00]

Gibbons: Exactly. Your son also, you're very close to him. He's a professor of computer science, just left Carnegie Mellon, is working at… Where is he in Chicago now?

Blum: He's in Toyota Technological Institute, which is located in the University of Chicago. The students… It's an institute that's concerned with machine learning. He's really a machine learning person. He's going to be hiring people in machine learning and he's already got a faculty of about 10. They get students from Chicago and also people apply directly. A lot of people in computer science, the young people by and large, want machine learning. That's really what they're after. So this should be a wonderful, wonderful opportunity for him.

Gibbons: I understand you're very lucky that you have a great relationship and you talk about work with each other and other things as well. Has he been an important person for you to bounce ideas off of and to learn from?

Blum: For sure. He points out to me when I'm going in the wrong direction, as Lenore does, often the same place. [laughs] I remember when he was… my son when he was six years old, realizing, "He can think. He can really think." I could ask him, "Do you think you can go on the bus by yourself?" That was at a time when no six-year-olds were going on buses by themselves. He thought about it and about the change that he had to make going from one bus to another, and he said, "Yeah, I think I can do it." I could see that he was really thinking well.

And I learnt computer science from him. When he was in sixth grade, his math teacher gave the class a problem, basically a theorem to prove, a theorem in number theory, and it was hard to see how to go about proving, hard to see why this theorem might or might not be true. What Avrim did is he took a little computer that I had built… I had built this little… This was before there were laptops. I had built a computer and a teletype, and he took this thing and he programmed it to get some insight into the problem, to get examples. He ran the thing all night and in the morning, he could see what was going on and he got a proof. I thought to myself, "That's really the right way to do mathematics. You do these experiments, you find out what's going on, so that you can prove the theorem." I learnt that from him. It's great. Sixth grade, so he was 12 years old at the time.

Gibbons: That's amazing.

Blum: Yeah, he's really… We have a very good relation. He's a very responsible person. I could always… He's responsible. I could be sure that he would do the right thing, whatever it is.

[Recorder was paused briefly]

Blum: What was computer science like?

Gibbons: When you were getting your PhD.

Blum: Or even before that. I mean when I was… Yeah. This was the time when… I guess it's 1958 more or less when we had a TX-0, and the TX-0 could run for about five minutes before there were errors. That was interesting. It was a vacuum-tube computer, and a vacuum tube would blow or the heat would get up too high, something would cause us to have to restart the computer. Somehow we had to figure out how to get answers when there were constant breaks with the wrong stuff. It's because of that in fact that McCulloch was so interested in figuring out how the brain could work despite the errors.

Gibbons: What was the first computer that you used?

Blum: The first one I used was actually… The first one I really used was probably the one I built when I was in Berkeley. Well, that's not completely true. At MIT, we had CTSS, Computer Time-Sharing System. That was kind of nice because you could have a… Well, it was equivalent to a laptop and you could be using the computer, and that was nice. Then we went to Berkeley, and Berkeley, when I wanted to use the computer, it was on cards. I had to sit down and make up cards. It seemed so old-fashioned by then. Then you'd put in your deck of cards and you wait a day for it to come back to tell you that there's a bug here. [laughs]

It's amazing that… I looked over my thesis, my PhD thesis, and I found that I had written it three times. There was the first draft, the second draft, and the final. When I think about how I actually write now, well, it's constantly rewriting, it's constantly fixing things up, which I don't even know how I could possibly have done… it was just three. Things were very different then.

But for me, yeah. And for me, building that first computer was a lot of fun. It was not a very powerful computer and it had a just very minimal computer language, but nevertheless, it was enough for Avrim to be able to program his math problem and find out what was going on.

Gibbons: So that was… you built it with him, this was the first one?

Blum: I built it and then he started to use it.

Gibbons: Interesting. After you wrote your thesis, at the time you were getting your first job, you said your thesis got you into Berkeley. How many computer science departments were there and why did you go into computer science with a PhD in math? Tell me about that.

Blum: Oh no, I really wanted… I never viewed myself as a mathematician. Lenore is a mathematician. I'm not a mathematician. I'm an electrical engineer who unfortunately doesn't know much electrical engineering and so does mathematics instead. But I'm very much an engineer, and computer science was just much more comfortable for me.

Gibbons: Were there many departments of computer science at that time? Where were they?

Blum: Yeah, they were starting up already. They were at Cornell and Stanford. There were some good places that had it. But it was just beginning around the time I got my PhD.

Gibbons: And how did you end up at Berkeley?

Blum: Yeah. How did I get a job at Berkeley? There was a professor, Lotfi Zadeh, who visited MIT. I had mentioned to Marvin that I would like to get a job at Berkeley, and so he introduced me to Lotfi Zadeh. Lotfi was encouraging and supportive and brought me over. That was very nice. That was time that my thesis was really considered very good, and so I was able to get a job. But unfortunately, as I may have mentioned, I kept on working in the direction of the thesis rather than looking around and going in the best direction.

Gibbons: How long did you work in the direction of the thesis? And tell us what you worked on there. Was this the recursion-theoretic attempt to do machine learning, or was that later?

Blum: Yeah, that came later, but that was still… that's an example, recursion-theoretic attempt to do machine learning.

I had some… Actually, I've had some good graduate students, but what people don't understand is that I probably learn more from my graduate students than they learn from me. For example, number theory. I love number theory. When I was associate chair for computer science, my time was completely taken, but I tried every morning for one hour to read some number theory. That was very important, because after three years of being chairman, my brain would have been dead except for the fact that I'd been reading this number theory. And the one who really got… one of the people who really got me into it is Len Adleman. He's a very good number theorist.

Gibbons: Also a Turing prize winner.

Blum: And he got a, yes, Turing prize for the RSA algorithm. You know how that came about, Rivest, Shamir, Adleman [2:00:00] for the RSA public-key encryption? The way it worked is Rivest and Shamir were trying to come up with this public-key cryptosystem, something that had been hinted at, really suggested by Whit Diffie and Marty Hellman, who also got a Turing Award later for their work. The way the RSA came about is R and S, Rivest and Shamir, would come up with some idea for doing this public-key and then they'd show it to Adleman, who would break it. [laughs] Then they'd do it again. They'd go back to the drawing board, come up with another idea, Adleman would break it. When finally Adleman couldn't break it, that's when they published. That was the RSA. It's interesting that to this day, nobody has broken it. I mean Adleman was really good. He really was terrific.

Gibbons: When you were doing the research, the inductive inference, you mentioned that Dana Angluin was a student?

Blum: Yeah. After Adleman, there was Dana Angluin. I was interested in…

Gibbons: What did you do there?

Blum: I wanted to do learning, machine learning, and I didn't quite know how to do it. Dana Angluin was there and we decided to try to do that, try to do machine learning. In fact, I wanted her to come up with a certain model, which she eventually did come up with, which was just wonderful. I wanted a model, sort of a teacher-student model, a theoretical model, which she constructed as you have the teacher wants to teach something to a student, so the teacher gives examples – "This is a dog, that's not a dog. This is a dog, that's not a…" – gives examples and then the student comes up with an algorithm for distinguishing dogs from non-dogs. Then, if it's not a good algorithm, the teacher comes up with a counterexample and shows it to the student.

This was the thing. This had been missing. This was a very important part of getting machine learning off the ground. You come up with this counterexample and then the student works at it and tries to come up with an algorithm that's better, and it goes back and forth like this. Doing this, basically the theorems are saying that you can, doing this, get a machine to learn.

Gibbons: Was this inductive inference?

Blum: This was inductive inference. One of the wonderful things… Dana Angluin is absolutely fantastic for many reasons. But one of the things she did is after she got her PhD, she went to work with Les Valiant, who's another Turing Award winner, in Edinburgh, Scotland. There she tried to get Les working on this, and eventually Les did buy in. He did not buy into the model. He bought into another way which he said would be much more powerful, and that's his PAC model, "probably almost correct" learning. And that was really good, but that came basically because Dana was pushing him to do that learning. So I'm kind of pleased with that.

Gibbons: Yeah. Adleman was one of your first students and then she came after him?

Blum: I guess, yeah, he was one of the first and she was roughly about that time too.

Gibbons: You have an amazing set of students. You have at least 35 PhDs who had over like 230 students of their own, and you've had three Turing prize winners among your students. I think Lenore would say you didn't just learn from them. There must be something that they're also getting from you. Do you remember your contribution to any of this work, those two in particular?

Blum: Well, maybe. The fact is that many of the students who did not get Turing Awards really should have. Kind of I would like to talk about them.

Gibbons: Yes.

Blum: I mean Dana Angluin is one who should have gotten some awards. Another one is Ronitt Rubinfeld. I would love for her to get the awards that she really deserves. She started working with me on program checking, which I won't go into, but basically… Well, okay, I have to go into it. Program checking where you write an algorithm to do something, and I was trying to convince people to check your work. You know, that's what you were taught when you went to school. You do some arithmetic, "Check your work." But nobody was saying *how* to check your work, especially when it came to writing programs.

So I did that with Ronitt. We looked at… You know what it means to sort a string of numbers. You have a set of numbers and sorting means to arrange them from the largest down to the smallest to order them. Normally this requires a lot of computation. This is an element takes $n \log n$ steps. Well, turns out the people who wrote these sorting algorithms – and in fact people are still writing these sorting algorithms – they invariably write into their program a checker. Namely, after their program has generated this ordered list, they run through and make sure it's really in order. Makes sense, right? One thing that was missing from that is you also have to check that the elements you get are the same that you started with. But that can also be done in essentially linear time. You can check and make sure that if there's an error, you'll discover it.

So people put in these checkers. Then, when they finally had their program written, they would take the checker out. I was trying to tell people, "Don't do that. Don't take it out." The sorting algorithm takes a lot of steps, $n \log n$ steps. The actual checking is just linear. It's unnoticeable – I don't know if that's a word – unnoticeable in the computation. Why take it out? Just leave it in. And I have on occasion with some programs I've written been surprised to suddenly discover that the checker screams that there's been an error where I would never have expected it.

This was basically what I was trying to do with Ronitt. And Ronitt went further in a very good direction. For checking, sometimes what you want to do is just sample the data to get the… to make sure that you're in the right ballpark. What Ronitt did, is said, "Look, this business of sampling can be you don't have to look at all the data, just a very small amount of it. You can work in sublinear time." So this whole field of sublinear computation is Ronitt's thesis, Ronitt's area. She's absolutely fantastic, and it's a wonderful field that she started there.

Gibbons: So you could look in a sub-area and it was a proxy for whole other areas of code, so you could check it for…?

Blum: So the idea, I guess part of checking is to make sure that your answer makes sense. That you're told, "Oh, make sure at least it makes sense." That's essentially what she was doing. She was coming up with a theory of how to make sure it makes sense, and in fact even sometimes how to do the computation you want. Once you can do that, you can do the computation sometimes without looking at all the data. That's her sublinear computation.

So actually it's the students that did not get these awards that I'm most interested in talking about. There's another. So Dana Angluin, Ronitt Rubinfeld. These are women. They really should get these awards. And there's another one at Carnegie Mellon, Mor Harchol-Balter. She's a wonderful computer scientist. She works enormously hard. You come in in the morning, she's at her desk working. There are students there all the time. You leave at 6 p.m., she's still there too late. No awards really.

Gibbons: What did she work on? What did you work on with her when she was your student?

Blum: Okay. I know very little about her stuff. She works on what's called… It's got a name. Sorry, I'm blanking. Performance evaluation.

Gibbons: Let me restate it. What did she work on? Wait, tell me her name so I can say it properly.

Blum: Yeah. Mor Harchol-Balter. [2:10:00] Mor Harchol. Married, Mor Harchol-Balter.

Gibbons: What did Mor Harchol-Balter work on and what was your part in that?

Blum: First of all, what she worked on is just stuff I learned from her. It did not go the other way. I learned from her. She worked on performance evaluation. In that area, performance evaluation, you're talking about a computer getting many jobs and you're interested in how to order the jobs or how to send the jobs to different computers so that you're getting the most work done in the smallest amount of time for the least cost. The models that were around at the time that everybody was using were the wrong models.

Gibbons: What year are we talking about roughly?

Blum: I was at Berkeley and she… I think 1990 to '95. The people were using really the wrong model.

Let me try to explain that. They were assuming… You know if you look at the words that are used in the English language and you draw which word is used the most, like "a" or "the," and then you make a list of the words in order of how often they're used, you come out with some sort of a curve. Then there's these words that you use very rarely at the end. People were assuming that that curve was an exponential curve. That means *e* to the minus *n*. The n^{th} element is appearing, the height of the curve is *e* to the minus *n*. It looks like this. *e* to the 0 would be 1, and then it keeps on.

This happens not to be the way… if you do this for actual words in English, you would find that this curve is wrong. The actual curve is one that was suggested by Zipf, Zipf's Law, and that says that this curve will have a polynomial expression. Instead of *e* to the minus *n*, it will be *n* to the minus 2 or *n* to the minus 3, which is a very different kind of curve. Instead of something like this, it would start the same way, but then it has what's called heavy-tail distribution. It just says up there for a very long time.

And none of the mathematicians were using that kind of curve because they didn't know how to do the mathematics. It was very interesting. It's sort of like that joke you hear about the person who loses their keys. "Why are you looking here under the…?" "Because the light is better here. I lost it there, but the light is better here." That's the way they were working. They were using the wrong curve because they knew how to do the mathematics. What Ronitt… Not Ronitt. What Mor did was to say, "Okay, look. We have to look at the correct mathematics. It's a heavy-tail distribution. Let's study it. Let's make the mathematics that we need to be able to understand data coming in according to that distribution."

This was wonderful. It's just a terrific new direction. And I was very supportive. I mean I really like the idea of doing it the right… looking where you lost the keys rather than under the light.

Gibbons: And this is her thesis? It started as part of her thesis?

Blum: That was her thesis. All the students tend to actually continue working on whatever their PhD was. As I did. I mean I did my PhD and then I kept working on that when I should have switched. But…

Gibbons: And why should you have switched?

Blum: Because my real heart is in consciousness. That's why I should have switched. It's really important to work on something you really want to know about. Because after all, whatever it is you're going to work on, you're going to work on it for the rest of your life. I mean that's at least my experience with these people. They start on something, they will work on it for the rest of life. Try to make sure it's something you're really interested in. I couldn't work on consciousness then, but I can now, so it's what I'm doing.

Let's see. I was talking about the students. Those are three woman students who really need to get some awards, actually. It's sort of a shame.

There's another person who really should get awards. That's Gary Miller. Gary is a number theorist par excellence. He was my student around the same time as Adleman and Dana. In fact, that's a lot of how I learned the number theory for computer science. Gary knew it. Gary explained it to Dana Angluin and that, Dana wrote it down where I could read it. That's how I learned the number theory that's right for computer science.

Gibbons: What did Gary work on as his PhD?

Blum: His PhD was number theory, and especially prime testing. He came up with a really nice algorithm for deciding if a number's prime. Nowadays, in cryptography, one needs to have large primes. It's wonderful that on my iPhone, I can test if a number is prime, a hundred-digit number is prime, and I'll get the answer in seconds. Won't take very long at all. The way I create a random prime is I just generate a random number and test if it's prime. If it's not, generate another random number, test if it's prime. If you do this, then in a number of steps about equal to the length of the number, that's how many times you have to try to find an actual prime. A hundred seconds is not too bad. If it takes a second a piece, a hundred seconds you've got a prime. So you can take a hundred-digit prime, you can generate it quickly, generate another hundred-digit prime, multiply them together, you get a 200-digit number. A 200-digit number, even today we don't know how to factor it in any reasonable amount of time. The mathematics is simply not able to. The computers are not yet even powerful enough to do that.

So Gary was really important because he produced this thing that enables us to generate these codes, produces the primes.

Gibbons: And that's important for cryptography and…?

Blum: For cryptography, for being able to send these secret messages.

One of the things I did at Berkeley was coin flipping, coin flipping over the telephone. This turns out to be really important, because in the kinds of protocols that come up between people, it's important for us to have the equivalent of a random coin flip that we can both trust. If we had the coin here, we could toss the coin and both of us agree on what it is. But unfortunately when computers are talking back and forth, one may be in New York, the other in California, and they have to agree on a coin toss. The question is how do you do that? How can two computers far apart agree on a random bit?

The reason one wants that is because these protocols generally need random numbers. They're just too slow without random numbers. You need random numbers. You also need them for the code itself. I've mentioned multiplying two primes together and then it's very hard to factor. So the way two computers can generate a random bit is one of them generates two primes, two large primes, multiplies them together, shows the result to the other one, and the other one has to guess… the largest has to guess what's the middle digit of the largest prime. Or if you want just a bit, what's the parity of the middle digit of the largest prime. The largest prime, there's a number there. 0, 2, 4, 6, those are even; 1, 3, 5, 7, those are odd. That's the parity.

So one generates the primes, multiplies them, sends the result to the other. The other one tries to guess. It can't factor the numbers, can't tell what that bit is, guesses. And then for proof that it came out the way he guessed, or not, he just gets those prime numbers. He's given those prime numbers, he can multiply them together to see that "Yes, that's the number I was given." He can check that they're prime – that's very fast – and now he knows the [2:20:00] result of that coin flip.

Gibbons: And this is a bit of handshake so they can work with each other without revealing their deep secrets? Is that the idea?

Blum: Right. In this particular case, one computer generated the primes and her deep secret is those primes. She doesn't reveal them. She just multiplies them and reveals the product. The other one has no way to factor it, can't get the answer. Only afterwards, after he guesses can he get those primes and check that he's correct.

Let's see. So…

Gibbons: That was cited in your Turing prize as an important contribution, and it's had impact on cryptography and… Tell me some more about its significance.

Blum: The significance is just that these protocols need to have these random numbers. So random numbers have come up a lot in the work I've done. This coin flipping is one way for two computers or two individuals that are far apart to generate a coin flip that they both agree on. But randomness comes up in many other, many algorithms. There are these randomizing algorithms which are able to get answers quickly that you can't get otherwise.

Gibbons: That sounds like some of the work you were doing in a couple other areas we need to cover. Finding the median of a set of numbers is something that you also did. Let's go to that and then we can come back to the pseudo-random functions, which sounded related too.

Blum: Yeah. The pseudo-… Yeah. The median is interesting because it's an example of a problem that is normally solved using randomness. You have a large set of numbers, and if you want to find the maximum of that set of numbers, you can do it quickly. You run through them keeping track of the largest number you've seen so far, and by the time you get to the end, you have in your hand the largest number in that set. You can find the max in just one sweep through the data. You can find the min the same way – just keep track of the smallest. Work your way through, one sweep, you will have the smallest. You can find the average of those elements in essentially the same way. You sweep through adding up all the numbers and at the same time count how many numbers you have. You take the ratio, that's the average. So it's kind of nice. You can get all of those in one sweep.

The question was can you get the median in one sweep? And there is a randomizing algorithm that can do that for you. It's a beautiful algorithm. The algorithm basically says, "Run through and pick a sample, a random sample of the numbers in the set." Just randomly sample. It's an example where you need to have random number, the ability to create random numbers in order to get a truly random set. Get a random set of numbers and then pick the median of that subset. That will be just an approximation to what you want, but once you have that, you can do one sweep to find out how close it is to the median. You might find that it's 10 bigger than, there's 10 elements between it and the median. Maybe it's 10 above the median, 10 elements in between. You want to find the right one, you do one sweep keeping a hold of the elements that are 10 smaller than it, and by the time you get done, you will have that element and the 10 just below it, and now you'll have the median in your hand. That's essentially linear time to find the median.

The question was could this be done without randomness at all? The median algorithm was intended to show how you could do that without randomness. It's still not… I doubt that it's really used, the median algorithm. If I had a program, I would use the randomizing algorithm. It's very clean, very fast, simple idea. The only trouble with it is that sometimes the number you get from the sample is just so big, maybe it's a thousand bigger than the median, and you can't hold all thousand. So there are occasions when it doesn't give you the answer, whereas the deterministic algorithm promises to give you the answer in a certain fixed amount of time, linear time.

That was the median algorithm. That's also an example where randomizing really is a great boon over the deterministic.

Gibbons: Does it help you solve it faster as well? Is that one of the significant aspects of it, if…?

Blum: Yes. In most cases, it's working to solve faster. In the case of Gary Miller's prime-testing algorithm, it really is a… he showed how to do it deterministically, but it basically showed how to do it as… it basically could be viewed as a randomizing algorithm. You can do it that way. He showed how to do it that way too. Then as a randomizing algorithm, it's what's actually used in our machines. I told you on my iPhone I can test if a hundred-digit number is prime by using Gary Miller's randomizing algorithm. It's very fast and there's a certain probability that it's wrong, a certain tiny probability. In fact, the way the iPhone does it is it tests if the number's prime and it runs the test five times to make sure it gets agreement, and then it tells you. I've never known it to fail. That seems to be just fine.

Gibbons: What is the name of that app? Do you know? [laughs]

Blum: Yeah. There are many apps that will do this, but the one I like the most is WolframAlpha. The reason I like WolframAlpha is that you don't have to know any programming language to do it. I can ask it, "Give me a hundred-digit prime, a random hundred-digit prime," and it will do it. Or I can give it a hundred-digit number. I could put on it the first hundred digits of pi, "Here's a number. Is this prime?" and boom, it tells you. And it uses Gary Miller's algorithm to do this.

Gibbons: Which was done as part of his PhD in…

Blum: That's his PhD, right.

Gibbons: …1960s? 1970s?

Blum: 1970s.

Gibbons: 1970s.

Blum: Yeah. I went to Berkeley in '68-69. Early '70s.

Gibbons: Okay, so early '70s. Interesting.

Blum: Early '70s. So he does this using number theory, which he taught to Dana Angluin, which they taught to me. So I really mean it when I say I learn at least as much from my students as they learn from me. Those are some of the students that haven't gotten awards and really should have gotten awards.

Gibbons: You also have here… Did we already talk about pseudo-random functions? How to get true randomness with a computer?

Blum: Uh, yeah, so…

Gibbons: It's related to this, but not exact. A little different.

Blum: Yeah. So yeah, the pseudo-random functions are interesting because they are functions, they are algorithms really that generate numbers. You put in a seed, small, short seed and it generates a very long string of numbers. This, these algorithms are based on numbers that are hard to factor. Basically you take a number that's a product of two large primes, you multiply them together, those two primes together, and you use that in a fairly simple way to take a short seed and generate a very large string of random numbers. Then the theorem says that if you can distinguish the numbers that this pseudo-random generator produces from truly random numbers, you can use that to factor. So these algorithms are based on these numbers. If you can distinguish, then you can use this to factor, and we know we can't factor. That's how we know you're not going to be able to distinguish. It's actually quite beautiful.

Gibbons: This I think has a role in cryptography also, or…?

Blum: Yes, a very strong role in cryptography. Yeah, so pseudo-random numbers come up a lot. I don't know. Did I show you? Lenore just a few days ago got a photograph from somebody who said, "I love your algorithm, your pseudo-random-generating algorithm, [2:30:01] and I've tattooed it." He's got a picture of the tattoo on his arm written in a programming language called Erlang. Lenore replied back, "Hmm. What does your mother think of this?" [laughs]

Gibbons: That was a Blum, Blum, and one of your students who wrote it.

Blum: And "Shub." Not a student, another…

Gibbons: Another colleague.

Blum: One of Lenore's co-authors.

Gibbons: Who was that third person?

Blum: Shub. Michael Shub. S-H-U-B. It is a beautiful algorithm and I should show you this picture with the tattoo. And he said his mother was okay with it. [laughs]

Gibbons: Non-interactive ZK proofs, zero-knowledge proofs. This was done with two of your students in particular, Shafi Goldwasser and Silvio Micali. I may be mispronouncing it.

Blum: Micali. Right, right, right. No…

Gibbons: Tell me about that, what that is and how that came about.

Blum: Sure. Well, they already have Turing Awards, so I don't want to speak a lot about them.

Gibbons: So just speak shortly. [laughs]

Blum: Speak shortly about them. [chuckles] Actually, there's something very important to say. Silvio, Silvio Micali, his father is a judge or was at the time a judge. Italian. He was a Sicilian judge. Can you imagine a judge in Sicily? He's still alive. Can you imagine that? Sicilian judges I understand are killed off all the time. I met the fella. Silvio brought him over, his mother and father, to meet me, and it was great because they speak no English and I speak no Italian, but we got along beautifully. We understood each other. You know you don't need to know the person's language to be able to communicate. And every so often, Silvio there, he would try to translate and his father would say, "No!" and I would say, "No!" [laughs] We're understanding each other. Silvio couldn't understand how we could talk to each other. It was great. A very bright guy.

I mention him because as a judge, law, language is really important, having good language, being able to say exactly what you really want to say correctly. This is what Silvio brought to the theory. For him, he pointed out to me, it's his definitions that have really made it. He had beautiful theorems, but it's his definitions that have been so important.

The definition of zero-knowledge proof, for example. These zero-knowledge proofs, they are proofs of theorems, mathematical theorems which have a very curious property. I can prove a theorem to you so that… They are randomized so that you are convinced with high probability the theorem is true and that I know the proof, but in such a way that you cannot turn around and convince anybody else. It's interesting. I convince you by being able to answer questions you put to me. Only a person who knows the proof of the theorem can answer those questions. Because I can answer them, I convince you that I do know the proof of the theorem. You cannot turn around and convince anybody else, because you'll be given different questions, new questions. You won't be able to answer those other questions.

So yeah, it's a beautiful theory and it does require randomization, again as we talked about, and they generally require us to be able to produce random numbers between us. That's part of it. And what you were talking about was this particular, zero-knowledge proofs, where in fact what I do is I just ship the proof to you and there's no communication, and it looks almost impossible to do because we have to have these random numbers and we have to have communication. The whole protocol calls for communication and the point was to somehow turn that into something that could be shipped to you.

Gibbons: Safer?

Blum: It could be shipped to you and then of course you could ship it to somebody else and they would have that proof. They would know *that* proof, but they wouldn't be able to produce a new proof. It's safely in the sense that what you get would not help you to understand a reasonable mathematical proof, would not help you to be able to create *another* mathematical proof.

Gibbons: And the point of this, is one sort of like a prover and one's a verifier? Is that an aspect of it?

Blum: Yeah, so…

Gibbons: Tell me how this…

Blum: I'm the prover who's going to prove the theorem to you and you're the verifier who wants to verify that the theorem is true. So those are the prover/verifier. What Micali and Goldwasser and Rackoff did was to really define prover – I mean the concept is there – the verifier, and those protocols, what it means to be a zero-knowledge proof.

It's a very interesting thing here. Zero-knowledge, the whole idea is "I will give you this proof and you will not be able to turn around and give it to anybody else. You won't get any knowledge from me." But it's not true that you won't get any knowledge from me. You are going to discover that, first of all, not just that the theorem is true but that I know a proof. So you're going to learn more. And in fact the amount of conversation, the amount of talk we have to do gives you a bound on the length of the actual proof. When we talk like this, you find out that the proof is this long, not longer. This is how long it is. If you want a proof, you don't have to go beyond this. It gives you information.

The interesting thing is he called it "zero-knowledge." It's wonderful because there's a clear definition of how to tell when the proof is zero-knowledge, and it's not really zero-knowledge, but it doesn't matter. He said, "This is zero-knowledge." He's defined it.

In fact, what I will probably end up doing with consciousness is something like that. I will have a model, we talked about this theater model; and there's the short-term memory, which is what we are conscious of; there's the long-term memory, the processors, long-term processors, which are looking at what's going on on the stage; and I will simply define that what's on the stage is conscious. That's it. I will define it and people will be very unhappy with it, as they were with the whole concept of zero-knowledge to start with. But then we can start to prove theorems. I'll say, "What's up there is in fact what we are conscious of," and you can start to realize, "Gee, when you look around, yes, you are conscious of what you see." Basically you are conscious of what you see, you're conscious of the inner voice talking to you, you are conscious of these feelings you get, like somebody… Those are the things you are conscious of, and no matter how long you hold that freezing-cold can of Coke in your hand, you will not be able to go and find out where that's coming from, where that agony is coming from. You cannot go back into the audience to see how it works.

Gibbons: So it's like a given for which you then can work…?

Blum: I will take that as a given. I will prove theorems from it. I'll explain free will. I'll explain or try to explain how one generates the agony of pain. And if I'm able to prove enough and to make it clear enough, then it will be accepted. At least that's how it worked for Silvio and that's how I'm hoping it will work for me.

Gibbons: Are there other ways it's being applied, to help solve difficult problems or…?

Blum: Yeah. Computer science tends to do this. Silvio is important because his father is a judge and he *knew* the importance of good language. So definitions for him are really important. But it really does… it does appear in other places where simply one defines mathematically, "This is the concept," and then uses a word that maybe is related to it.

Gibbons: Then you also have been involved with the reverse Turing test, the CAPTCHA.

Blum: Oh yeah. [laughs]

Gibbons: Which you can tell me what that stands for and who you worked with on that. Let's talk about that fun project.

Blum: Oh yeah. I made up that word "CAPTCHA." This is these funny-looking squiggles where you have to say what the letters are, or the digits. It took me a while to come up with "CAPTCHA." I was looking for a "gotcha"-like word, and "CAPTCHA" is wonderful. It stands… "C" is for "Computer"… [2:40:00] No, "Completely." So "Completely Automatic Public Turing Test to Tell" – it's four "T's" – "Computers/Humans Apart." C-A-P-T-T-T-C-H-A. [chuckles] I was very proud of that.

Yeah, the way that came about was actually very good. It's a good way to do research. I had asked Udi Manber, who was chief scientist at Yahoo! at the time, to talk to the faculty at CMU about some problems that came up in Yahoo! You know, we're theoretical computer scientists. We like to hear problems that they have and sometimes we can solve them. He mentioned three problems, one of which was the chatroom problem. The chatroom problem is Yahoo! was being bothered by… well, people in chatrooms were being bothered by bots coming in and talking to the people. In fact, Lenore immediately afterwards realized this because she wanted to buy a laptop, a new laptop, she said what she wanted, and she went to a chatroom to say what she wanted and somebody came back, "I know exactly what you want," and what it was was an advertisement. See, it was just a pointer to a certain advertisement which actually didn't have anything to do with what she wanted. That's the problem. The bots come in and they act as if they know, and they disrupt what's going on. But yeah, what Udi Manber wanted was a program that would be able to test the people that are coming in and be able to weed out the bots. Just let humans into the chatroom and keep out the bots.

This is a very strange sort of problem. It's strange in the following sense. Basically what he's asking is for some sort of… he wants a computer test. He wanted an algorithm. He wants a computer test that should be able to… an algorithm that should be able to generate tests for testing these different people, and it should be able to tell which ones are human and which ones are not, because only the humans should be able to pass the test. Yet this algorithm itself, being a computer, should not be able to pass the test. It itself can't pass the test that it grades. So the puzzle is "Can you write a program that will be able to grade somebody on a test that it itself cannot pass?" And we professors know that it's possible.

Gibbons: [laughs]

Blum: [chuckles] But the students don't always realize that.

Gibbons: You said you had a challenge finding something too that humans could do that computers could not do.

Blum: It was really hard. We kept looking and looking. For example, we tried IQ tests. After all, an IQ test measures your intelligence. You can find IQ tests and solutions, and it turns out that computers can pass those IQ tests better than any human can. They're faster and they're perfectly competent. We tried one thing after another like those IQ tests. You know, algebra problems, all sorts of problems. Very hard. We must have worked on it for a year trying to come up with something.

Then I was on a trip to Berkeley and I spoke to Dick Fateman, who I told the problem to as I told to others. He said, "I know just what you need." He said, "I've been teaching a course on OCR, optical character recognition, and these computers cannot read handwriting or typewriter-written stuff. They're just not as good as humans at reading it. So make that your test." And it was a great idea, because in fact kids, very young kids can already read stuff that computers cannot read. And I wondered about that.

So here's the test. It sort of explains, I mean what the computer has to do is it takes some legible… it generates some characters, which of course it can read. It prints them, it knows what they are. Then it twists them to the point where it cannot read them, that's what it gives out, and the human can read them. So the human is able to read what it, the computer itself cannot read. The computer can grade it because it generated the letters in the first place. It knew what letters it had before it twisted it, so it knows what the answer should be.

So that took care of that, but I asked myself, "How come kids can do this? How come kids can read stuff that's so hard for a computer to read?" I realized after a while that very young kids sit at their mother's side, and their mother's reading to them from a kids' book that's sort of floppy in strange light, who knows, and she's reading and the kid's looking from the side, not even from in front, and is learning how to read that way. So the kid is really being given exactly this kind of mangled information and learns to read. Then the kid goes out in the car and they pass signs on the street. If there's a stop sign, the kid will say, "Ah, that's 'Stop,' " even though there's bird doo on the stop sign and there are shadows covering it. But still the kid can read it. So even very young kids can pass this test, and computers have a very hard time with it.

The other thing I like about this is that there's another angle to this. People say, "You know, at some point maybe computers will be able to read this," and I tell them, "Yes, I want them to. I want them to be able to read this. We'll have much better optical character recognition when they can." So the wonderful thing about these tests is that we give them as challenges to people who are in vision or optical character recognition or whatever. They are challenges for them to come up with a program that can read the stuff.

So we provide these challenges and then we had a professor at Berkeley send us, "I have written a program to pass this" what we called "easy CAPTCHA." The way we can tell that he did in fact write such a program, is we generate 5,000 CAPTCHAs, we send them to him, and a minute later he gives us the answers. And we know that no human being could do that. He really did write a program to do this. So my hope actually is that eventually these CAPTCHAs will fail, that eventually computers will be able to pass all CAPTCHAs, because then we'll have good optical character recognition.

**Gibbons:	So using the CAPTCHAs to train the computers in this pattern recognition or…?**

Blum: Yes. Or at least to check on how well the computers are doing. When we first started with this, we weren't sure where the programs for breaking CAPTCHAs would come from.

Gibbons: Stop one minute. Maybe we can ask him not to… Do we take a break?

[Recorder is paused for a break]

Gibbons: So tell us about reCAPTCHA. What is that and who did you work with on that?

Blum: Well, all the work on CAPTCHA was with Luis, and we did talk about reCAPTCHA also, but Luis really pushed that to actually find a way to make money off these CAPTCHAs.

Gibbons: Luis von Ahn.

Blum: Luis von Ahn. The way he made money off of it is that *The New York Times* wanted to be able to digitize the old *New York Times* that were printed using this type that sometimes doesn't get the letters in exactly the same place and sometimes it's smudged, and these are sheets of paper, I saw them, where somebody had written on them. And OCR could read some of this, but not all of it. So they would take the ones that they can't read and give them to Luis. What Luis would do is make a CAPTCHA by taking a word that OCR could not read and a word that we did know how to read, and he put the two together and he'd give it to the bot or human, and require them to actually read both. If they could read the word [2:50:00] that the CAPTCHA knew the answer to, then it was a human. Then we would accept what it gave for the other word tentatively as being the right answer. You give it to several people and if they agree on what that second word is, that's it, that's what *The New York Times* is looking for.

Gibbons: They did that word-by-word for all *The New York Times*?

Blum: For every single word that OCR could not read. It's not all of *The New York Times*, it's just the parts it could not read.

[Recorder is paused briefly]

Blum: Oh, yeah. Something like a penny a piece. The words were half a penny a piece. [laughs]

Gibbons: That's amazing.

Blum: So he started to make money with that. That was good. Luis is my first entrepreneur ever, really. He's been very good at it.

And I use his Duolingo every day. His Duolingo is this wonderful program for learning languages, and I use it for two things. I use it for Spanish, which I know quite well, and even though I know it very well, I'm still learning. It checks my pronunciation. It has improved my pronunciation. It teaches me words I don't know. It's improved my grammar. My Spanish was very good, but I'm still learning from it. And I use it for German. Now German was my first language, but it's completely gone. But, you know, I sort of hold it in my heart because my parents spoke it. So I decided I would like to learn German again. So I'm a beginner at that and I do a lesson in German every day, and you know, I'm learning German. I really am learning German. It says I'm 47% fluent now. *Ich kann ein bisschen Deutsch sprechen*. [laughs]

Gibbons: [laughs] This is your student's… This is your intellectual grand-project in a way, second…

Blum: This is Luis von Ahn's. No, so Luis…

Gibbons: He did this on his own though.

Blum: This is his program, yeah. And in fact it's his company. It's a company now with some 60 people working for him. A lot of people. And they're trying to get as many languages as they can. They are using it to teach English to Chinese students. It's not yet at the point where they can teach Chinese to English users. In fact, I just tried yesterday to see if I could start to learn Chinese from it. No, it's going to teach English to Chinese users, not the other way around.

Gibbons: What is so difficult about Chinese, I wonder? Is it the…

Blum: Well, you know, it's a tonal language for one thing, and the writing is very, very different. I spent a couple of years in Hong Kong while Lenore was writing her book. She wrote a book with Smale and Shub and Cucker, a very good book. She was there to work with these co-authors. I decided I would learn Chinese. I wanted to see if my brain still works.

Gibbons: Does it?

Blum: [laughs] So I spent two and a half years on Chinese and I can proudly say I'm at the point where I'm as good as a two-and-a-half-year-old. [laughs] It's an interesting language. It's tonal, so I had to learn what that means. I taught in English when I was in Hong Kong an operating systems course, and my first sentence to the students was in Cantonese, very slowly. The sentence was "*[speaking in Chinese]* guǎngdōng huà." "Guǎngdōng huà" is Cantonese. When I finished with that, they clapped. [laughs] It took me a long time to say it, then I told them, "I'll teach you operating systems. You're going to teach me Cantonese." That's the agreement we made. Very nice.

Gibbons: A little bit like CAPTCHA. [laughs]

Blum: [laughs]

Gibbons: Worked on two ways. Do you have anything more you want to say about CAPTCHA or reCAPTCHA? Have we covered most of that now?

Blum: Yeah, I think we've…

Gibbons: That we've covered that.

Blum: …covered that.

Gibbons: How about students? You have had such a rich legacy of students. We've talked about quite a few. Are there any others? I think you mentioned Eric Bach.

Blum: Yeah. I'm especially interested in talking about the ones that did *not* get awards yet and what I learned from them. What Eric Bach did for me was to teach me what a number looks like when you factor it, what a random number looks like when you factor it. He has a wonderful image of it. Namely, you take this number. It's this long, a long number. Essentially, the way the random number looks like, he describes this as a dartboard. You throw a dart along the length, and wherever it falls, you then fill in the first part with the prime. Then in what's left, you throw a dart, and wherever it falls, you fill in what's left with a prime. And you keep on going. This can be made precise and it is a beautiful, beautiful answer to the question "What do numbers look like, random numbers look like when you factor them?"

And you can get theorems out of them. For example, if it's an *n*-digit number, how many darts do you have to throw? Well, the first one will fall more or less in the middle. The next one will fall in the middle of what's left over, etc. So the number is *n* digits long, it'll take about log *n* dart throws, about log *n* primes. And that's a theorem. Yeah, a random number has expected log *n* primes. And in fact expected one roughly half the size, the next a quarter the size, and so on.

There are lots of theorems like this that come out of this beautiful dartboard explanation that Eric gave, and for which he has never really gotten the credit he deserves. It's such a beautiful intuition that he was able to give me, and I needed it for the cryptography and the number theory.

Gibbons: That's fantastic. You have had many good students and many good people you're working with. Do you have any particular ones you're particularly proud of?

Blum: Well, you know, I'm proud of these women. I didn't mention Shafi Goldwasser, who's another woman. I think these women have been spectacular.

Gibbons: It's great you've been a mentor to them. You talked about what important mentors you had in your career, and I'm curious if there's any other… You've obviously emulated some of the things that were good that they gave to you, such as listening carefully and encouraging your students.

Blum: Yeah, my mentors…

Gibbons: Your mentors did that for you and you seem to have wanted to do that yourself.

Blum: …listening, and they were supporting me. Right. They were really very supportive. I mean Warren would tell me, "Oh, your IQ is 160." "Well, [laughs] what do you mean, my IQ is…?" He said, "You're way above average. Way above." You know, it didn't matter whether it was true or not. It was so uplifting, it was so encouraging, and I find the sa-… I find actually though that these students really have great ideas, and it's a pity that sometimes they come up with these great ideas and they're rejected.

There was a fellow that came up with this whole concept of public-key cryptography named Ralph Merkle. He was a graduate student in Berkeley when I was there. He had a wonderful idea for creating these cryptosystems. He talked about creating a puzzle sort of like a jigsaw puzzle. You can create a jigsaw puzzle by taking a square and then drawing lines where you're going to cut, and you now know the answer because you drew the lines. But you can give it to somebody else and if it's mixed up, they may find it very hard to get back the answer. This can be used for cryptography. For example, it could be your password. You can say, "I know how to turn this into a… I know how to make these jigsaw pieces come together." That could be your password. But it's useful for a lot more than that.

Anyway, Ralph Merkle presented his idea to Lance Hoffman. And Lance Hoffman, good guy, works in cryptography also, and if you read the letter he got back, "Nah. This other idea is much better. This will never work. Forget it." And it's such a shame because this was a great idea. The point I guess is that their ideas… if they come to you with an idea, they're good people, they come to you with an idea, they will have thought about it, [3:00:00] it's probably a much better idea than you realize, and you should listen to it and be encouraging of it.

Gibbons: You said you also tell them sometimes that they're going to become the world's experts.

Blum: Oh, right. Students want to know what I tell these young people when they come to me. I guess the ones I accept are the ones that come to me with some idea that really thrills me. They come to me with an idea that thrills me, then I'll take them on. What I have told people, well, entering graduate students who are going to get their PhD, what I tell them is "It's going to take you a long time to get your PhD. You'll be working four or five years to get it. It will be a lot of hard work and you will feel like you're working on some tiny thing. You have to realize first of all that you're going to become the world's expert in that thing. It will be an important thing – that's why you're working on it – and when people need to know about it, they will come to you as the expert in that thing." It reminds me of Blake's poem "To see a World in a Grain of Sand," because that grain of sand is what they're working on. And the more you work on it, the more you will see the entire world in that grain of sand.

Gibbons: It's a long way from the advice you got at P.S. 86. [laughs]

Blum: [laughs]

Gibbons: That elementary school teacher.

Blum: "You'll never make it."

Gibbons: "You'll never make it," yeah.

Blum: Yeah. Right.

Gibbons: So I think the last question that we haven't asked you is they want to know how you reacted or your family reacted to the Turing Award. What has that meant to you?

Blum: It's meant a lot to me, because it's kind of scary actually to work on something like consciousness. It's scary because at one time it wasn't permitted, and nowadays the neuroscientists really know some wonderful things about it and I'm very excited to find out about this. But the truth is that in computer science, it's probably a very… a lot of people turn up their eyes when I say, "I'm really trying to build a conscious machine." They just don't… Mostly they turn up their eyes because they don't realize how much more is known than when they were young. A lot more is known and I do believe that this is stuff that can be turned into really good mathematics.

So it meant a lot to me because it allows me to work on something that… It's sort of like getting tenure. You get tenure, it allows you to work on stuff for longer or stuff that may or may not work. And getting a Turing Award is tenure magnified. You can really work on what you want.

Gibbons: That's fabulous. Can give you also some more confidence perhaps. If you approach somebody in another field, they know you're a Turing prize winner.

Blum: It helps, it helps, it helps. [laughs] It helps.

Gibbons: Absolutely. Well, I think we've… Anything else that you would like to add? Or anything…

Blum: I think you've done a great job of covering all this.

Gibbons: Thank you.

Blum: Thank you very much for doing this, Ann. I mean you've been really wonderful. Your questions are terrific. Whenever I stumble, you're right in there to help me out. I really appreciate it.

Gibbons: Thank you. It's fascinating.

Blum: It's been wonderful.

Gibbons: It's really been fun for me too. I wish I could turn this into a profile for *Science* if they'd let me write about computer science. [laughs] I can't. It's…

Blum: No, you keep on writing about what you're writing, because that's the stuff I like to read about.

Gibbons: Yeah. But it's just wonderful. Yeah. I find it very inspiring to talk to people about their careers, and yours has been such a creative journey. I mean what a wonderful journey. Especially after that start, it's rather moving that you overcame lower expectations, although maybe not from your parents.

Okay, one more question. Are we still rolling? And they'd have to cut all that stuff out. Your parents, how did they feel? You did become an engineer, but did they see you become a professor and how did they feel about that?

Blum: First of all, none of their sons became what they wanted them to become. [laughs] So that's one thing. I think my father at the end accepted the fact that I was a professor. He did come around. So that was nice, because he was dead-set against me doing something like that, where you wouldn't be able to make a living.

Gibbons: Did he catch the excitement? Did he feel proud at the end?

Blum: [laughs] Well, I'll tell you this. He decided to go to college when he was my age. He was about 80 years old. He went to college in New York, Fordham College, and it was fun to go there and to see this classroom of students, normal students like I would have, and then they would come out of the classroom. I definitely wouldn't see him. He was in the middle and he was shorter than all of them. [laughs] And it was impossible to see him, but there they were, and I was so proud of him.

Gibbons: That's wonderful. Well, thank you.

[3:05:32]

[end of recording]