

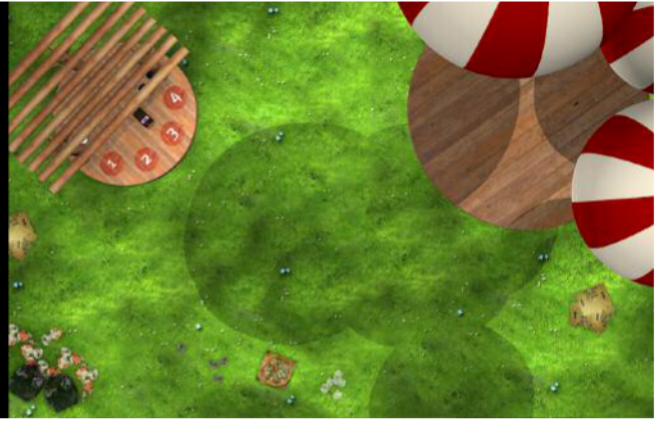


Christmas Lectures 2008

Hi-tech Trek
Presented by Professor Chris Bishop

Lecture 3: The ghost in the machine

Broadcast on: 31st December 2008



Part 1
Professor
Chris Bishop

Did you know that computers are the most versatile machines ever invented?

Have you ever wondered how they can store thousands of songs in your pocket, or how that invisible thing called software really works?

Follow me as we track down the Ghost in the machine.

Hello and welcome to the Ghost in the machine. Computers are extraordinary things, they can simulate virtual worlds with impressive realism. They can play chess better than almost any human. They can even tell us our position, anywhere on the planet, to within a few metres. So is there no limit to the capabilities of computers? Well perhaps there is. Later, we'll see how a simple jigsaw puzzle, with just a few dozen pieces, might defeat the world's most powerful computer.

So how are computers able to do these extraordinary things and why do they also have limitations? Well to find out, we're going to explore something very special in this lecture. You can't see it and you can't touch it, but without it the digital revolution would never have happened. To help us understand what this is, please welcome astronaut Stefan Gates.

Stefan, welcome, what is it that you're making?

Stefan

Ah, well, ah, we're going to make, em, an ice cream, it's going to be the world's most adventurous ice cream. Ah, we start off with one of these, this is a crème anglaise, ah, which is basically custard to you and me. Em, and that's a bit of, ah, egg, milk and cream. Now we add to that some lemon grass – very fragrant, very nice – and some lime zest – very zesty. But not very adventurous yet, so I thought what we'd add is a poisonous, bottom-dwelling sea monster. This is jellyfish, really tasty, and not actually poisonous this one, it's a special one that's, ah, used for, ah, cooking in China. We're going to add that in and then, to justify my wearing a white coat and goggles, we're going to freeze it in the quickest method on the planet – this is by adding liquid nitrogen in at -196° centigrade.

Okay, Lisa, let's go.

Chris

Okay, that's going to take a moment, so while we're waiting, we can think of this as a bit like a computer.

Stefan has utensils, ah, the hardware, they're like the processor and the hard disk of the computer.

But there's something else that's essential, and that's the recipe. Now Stefan has kindly given me the recipe for jellyfish ice cream and here it is. The first step is to mix the lime and the chopped jellyfish, and so on, into the mixture. The second step is to add liquid nitrogen and stir – it looks spectacular. Okay, and the third step is to test to see if it's cold enough. If it is cold enough we stop, because we're finished. If it's not cold enough, we go back to step two, add more liquid nitrogen and stir and then we test again in step three.

Okay, looks like it's ready. I'm dying to try some.

Stefan

There we are.

Chris

Fantastic. It's still hissing a bit, is that okay?

Stefan

Ah, no it's not, you're just being a little bit scaredy-cat.

Chris

Oh no, I'm not a scaredy-cat. Here we go, mm, mmm, it's very unusual, it's extremely – it's actually very good, it's extremely good. Who wants to try some, who's brave enough to try some? You were first, go on, you have a little go. How's that?

Not bad? Not bad at all, pretty good, huh? Okay, good.

So a recipe is a series of instructions and a computer also needs a series of instructions in order to process data. Now together the recipe and the data from the software. Of course, this cooking recipe has just three steps, but the recipe for a computer would be much more complex, it would have millions of steps.

Stefan, thank you very much for joining us.

Stefan

Thank you.

Chris

The recipes that computers use are called algorithms and to help explain this, I'm going to need five volunteers. I think we have five people already lined up, so if you five would like to come on down.

If you'd like to just come and stand, just stand there, form a line, that's good, if you just bunch up nice and close to each other. Excellent. Okay.

Now, I have here some cards with letters on. I'd just like you to hold those up in front of you, like that, just nice and high. That's good, there we go.

Okay, so what we have here are the letters A to E, but they're all jumbled up. Now something that computers do all the time is to sort things. For instance, if you bring up a list of your friends on your mobile phone, then you'd probably like them in alphabetical order. So I'm going to show you a simple computational recipe that will sort these letters into the right order. So here it is.

So the first step is to start with the item that's on your left, that's the one at this end. The next step is to take each item in turn, down the line, and compare it with the next item on the right and if they're not in the right

order, we're going to swap them. And the third step is to ask, are all the items in order? If they are, we're finished and we stop, if they're not, then we go back to step one and we repeat it all again. Okay, so let's give that a try.

So step one, we start with the item on the left, we compare with the one on the right. Now those are in the correct order, so we'll leave those where they are. Let's look at the next two items, are they in the right order? No they're not. So what we'll do is I'd like you two to swap places please. Okay, that's good. We compare the next items, they're not in the right order either, so I'd like you two to swap places. And finally, these two, they're not in the right order, do you two need to swap places as well. Okay, that's excellent.

So that takes us on to step three – are they all in the right order?

Well no, they're not, so we go back up to step one and do it all again. So let's compare you two, you're not in the right order, so you two can swap places. And yes, you're in the right order, that's good. You two are not in the right place, so if you two could swap. And you two look good.

So we go back now to step three – are they all in the right order?

Yes, they are, so we're done.

Okay, excellent, thanks very much, you can take your letters back to your seats with you.

So software is based on recipes or algorithms, but how are software and data represented inside the computer? Well I'd like you all to shout out the answer to the following question. How far can you count on the fingers and thumbs of two hands?

Audience

Ten.

Chris

Ten. Okay, most people said ten there I think.

I'm going to show you a really simple, but really powerful way of counting, which would allow you to count to over a thousand on the fingers and thumbs of two hands.

And for this I need four people, who I think are already, ah, picked out, who are sat there. Who are the four people we picked out?

That's it, you come on, come on down.

If you'd like to form a line along there, that's it, just stand in a line. Just move slightly that way, that's it, just keep bunched up close together, that's good.

Okay, so I'm going to give you some cards to hold. Ah, that's yours, you hold that one there, that's yours, that's yours and that's yours. Okay, and you notice each card has twice as many dots as the card on the right. Now each of you is going to be a binary digit, or a bit. And in this powerful way of counting, each bit is either on or off. So if I could just borrow that for a moment. When you're on, that means your dots are facing forward and when you're off, that means the black side is facing forward. Okay? And if I tap you on the shoulder at any point, I want you just to turn your cards over, okay? And if I tap you on the shoulder again, turn the cards over. Okay, all got that? Okay, there we go.

So, at the moment you're all on and we can work out what that represents by just adding up the dots, so we've got 8 plus 4 is 12, plus 2 is 14 and 1 is 15. So that's binary for 15.

Okay, let me try tapping you and you on the shoulder. Good. So this is now on, off, off, on, which is 8 plus 1, that's binary for 9.

Now as well as using on and off, we sometimes also use one and nought, so this could also be written as 1001.

So what does that actually mean? Well it means that we have one lot of 8, no lots of 4, no lots of 2 and one lot of 1. Now if that sounds a bit strange, it's actually just like ordinary decimal numbers, so here's a, here's a decimal number. This is 2037, and what that means is that I have two lots of 1000, no lots of 100, three lots of 10 and seven lots of 1.

Okay, let's see if we can, ah, count in binary now. So I'm going to tap you on the shoulder, and you, so you're all off at the moment, that's zero.

And to get a 1, we do that, and that's a 2, excellent, that's a 3, that's 4, 5, 6, 7, 8, 9, 10, ah, 11, right, okay, you were ahead of me there, 12, good, ah, 13, yes, keep going, 14 and 15. Is that right? Yes. Excellent, well done, thank you very much.

Just take your cards back to your seat.

Okay, that was excellent. So there we had just four bits. Now computers usually work with groups of eight bits and we call that a byte. Now because four bits is half a byte, we call that a nibble. Okay.

Now each time we add an extra bit, we can count twice as far. So if somebody asks you how far can you count on the fingers and thumbs of two hands, the answer's not ten – with ten bits you can count all the way from 0 up to 1,023.

Now, let's suppose that everybody in these three rows of the lecture theatre were a binary digit or a bit. Well together, you'd be able to count to over 2,000,010.

What about if everybody in the lecture theatre were a binary digit? There are just over 300 of you.

Well, this is how far we could count with 300 bits. Here it is, it's a big number – here are the thousands, those are the millions, ah, billions, trill, trillions, um, keep going, my goodness, okay, this is a, this is a seriously big number. This number has 90 digits. In fact, in fact this number is too big to handle, obviously.

This number is bigger than the number of atoms in the universe and that's with just 300 bits.

So we've seen that computers follow recipes or algorithms and we've seen how numbers can be represented as binary digits, or bits. But to build a computer, we actually need to store the data inside the computer in some way. And to find out how we can do that, join me after the break.

Part 2 Chris

Most computers store their data on a hard disk. We can see how a hard disk works by looking at this disk here, which has a transparent case. Inside is the disk and, as I switch it on, you'll see the disk begin to spin up.

When that's up to speed, it's going round 10,000 times a minute.

You'll also see there's a little arm here, in a moment that will start flitting backwards and forwards across the surface of the disk. And it's that movement that you can often hear making the clicking sound from your computer.

Now to see how that works we have a giant model here. This is the disk and on the surface of the disk, we have lots of little magnets, and these represent the binary digits.

So here we can see some alternate blue and red dots. The red dots are magnets with the north pole up, those represent the ones, and the blue dots are magnets with the south pole up, and those represent the noughts.

Now this is that head that we've just seen and at the end of the head is a little sensor which detects the magnetic fields. And that's sending it's signals to this computer and you can see, indeed, that those little magnets in that inner curved track there are alternate ones and noughts.

I've got a real magnetic disk here, mounted underneath this microscope.

And on top of the microscope we've got a camera and if we can take a feed from that camera, please, and just bring it up on the screen, here you can see these curved tracks. And if we just, ah, zoom in on that – this here, this is a speck of dust, gives you some idea of the scale of this – these are the curved tracks and you can see they have these stripes. Each of these stripes is a little magnet, it stores one binary digit.

So we've seen how to represent numbers as binary and we've seen how binary numbers are stored on a hard disk. But what about other kinds of information? For instance, have you ever wondered how we store music on a portable music player?

Well, to find out, please welcome the person who composed the music for this year's lectures, Max de Wardener.

Beautiful.

Well while Max was playing there, I was recording the sound on this computer and I'll just play it back and we'll listen to that again.

Okay, now this yellow curve that you can see here represents that sound, it's the pressure waves in the air. And these pressure waves are being sampled 44,000 times a second.

If I just slide this cursor along, we can look at some of these samples. Here's one, it has a value of, ah, 31,000 and that's been converted into this binary number with 16 bits. I go along a little bit further, and there's some different value, that's 34,000. Again, that's been turned into a 16 bit number. So 44,000 samples a second, each of which is 16 bits.

But what about other forms of information, for example, images or video, or text, or even computer programmes? Well everything that's stored in your computer is stored as binary. Max, thank you for joining us.

Now the problem with storing music directly in this way is that it would take up a huge amount of space, just 10 seconds of music would need 14 million bits of data. If we want to fit thousands of songs onto a portable music player, we're going to need to find a way to compress the data. That means we have to represent the same music in fewer bits. Well, how can we do this?

Okay, for this I'm going to need a volunteer please. Who would like to come on out?

Yes, why don't you come out?

Can you stand here? Good. And, ah, what's your name?

Nikki My name's Nikki.

Chris Nikki? Alright Nikki, I want you to take this pen and in a moment, I'm going to ask you to think of a number, any number you like, between 0 and 31, okay, and then write it in nice, big writing on this board and then just hold it up, right way up, so that the camera can see it. Okay? I'm going to be blindfolded, so I won't know what the number is, and I'm going to try and work it out. Okay. So somebody's going to put a blindfold on me now and, ah, once I'm blindfolded, I just want you to think of a number, any number you like between 0 and 31 and just write it in nice big writing on that sheet for me, and then hold it up to the camera, so that everybody can see except for me.

Okay, have you done that?

Nikki Yep.

Chris Okay, and I want you to remember that number – want everybody to remember that number – and then I want you to turn the board over so that I can't see it, and let me know when you're ready.

Okay, you ready?

Nikki Yep.

Chris Okay, I'm going to take the blindfold off then.

Okay, so everybody knows the number written on that board except for me. I'm going to try to work out what your number is, okay, and I'm going to ask you questions. When I ask you a question, all I want you to do is answer yes, or no, okay, and people can help you if, if you get stuck. Right, my first question is – the number you've chosen, is it 16 or more?

Nikki It's more than 16.

Chris More? It's 16 or more. Okay, so the answer is yes. So I'm going to put up a 1 here, just to remind me that you answered yes. So I know the number is 16 or more. So my next question – is the number 24 or more, yes or no?

Nikki No.

Chris No, it's not 24 or more. Okay, right. So let's think. My next question is – is your number 20 or more?

Nikki Yes.

Chris Yes. Okay. Is your number 22 or more?

Nikki Yes.

Chris Yes, yes, happy with that. Okay. Is your number 23?

Nikki No.

Chris No, so your number must be 22 – is that right?

Nikki Yes.

Chris Okay, good.

Do you want to turn it over and we'll, we'll see – oh, it's up on the screen. Okay, so 22. So I managed to work out your number with just five questions and in fact, no matter what number you had chosen, between 0 and 31, I could always work it out in just five questions.

And the reason for that is because, all I was doing was working out the binary form of your number, like that. Alright, thank you very much, you can sit down. Thank you.

So my first question there was really – is that number in the top half? Is it between 16 and 31? Is it in the top half of the range, yes or no? And once I knew that it was, I then took that half and halved it again, I said, was it in the top half of that range? So this technique of taking a problem and then cutting it in half and then cutting in half again is very efficient, it's called divide and conquer and it's a trick that computer scientists use all the time.

Now I could have solved that problem in a different way. I could have asked 32 questions of the form – Is it 0? Is it 1? Is it 2? And so on. And the answer would then have had 32 bits. But I was, I was able to, to represent the same number, the same answer, using just five bits. So when we take information using a lot of bits and we reduce it down to fewer bits, that's called data compression.

Now music is often compressed with something called MP3.

And MP3 goes even further, because it throws away aspects of the music which the human ear finds very hard to detect.

Now taking music and compressing it, using MP3, is another example of a computational recipe or algorithm.

And there are lots of other problems that are much more complicated which computers are also very good at. So, for example, classical physics is very well understood and it's described by equations that were worked out over 300 years ago by Isaac Newton. Now I'm going to prove just how predictable and reliable these equations are by conducting an experiment, in which even a tiny variation in the laws of physics, could result in my getting killed.

I have here a solid steel ball – it weighs 14kg, it's incredibly heavy – and it's suspended from the roof of the Faraday Lecture Theatre by this steel cable. Now what I'm going to do is to take this steel ball over here and I'm going to stand with my back against this headrest and in a moment, I'm going to place it against my face and then I'm going to let go.

It's going to swing out across the lecture theatre and then it's going to swing back towards my face.

Now according to the laws of physics, it should stop just before it touches me. Okay, that's the theory, let's see what happens. I think this is probably worth a countdown, isn't it? Okay, are you ready? Three, two, one, go.

I'm, I'm very pleased that the laws of physics are nice and robust inside the Royal Institution. Whatever you do, of course, please don't try that experiment at home.

So the laws of classical physics are very well defined and computers can simulate these laws very accurately. For example, it's used in flight simulators. Flight simulators today are so accurate that an airline pilot that, who's training on a new type of aircraft, can actually do all of their training on the flight simulator. The first time they ever fly a real aircraft of that type, it's with fare paying passengers.

Of course software is also used to tackle much harder problems. And to give us his thoughts on the power of software, joining us live by satellite, from the USA, please welcome Bill Gates.

Bill, welcome to the Christmas Lectures and thank you very much for joining us.

Bill It's great to be here.

Chris Bill, we have a question from, ah, Thomas, who's sitting in the audience here. Thomas, do you want to ask Bill your question?

Thomas What do you think computing will be like in 50 years, when I'm your age.

Bill Well 50 years is, ah, a very long time. Ah, if we look 50 years ago, computers were so expensive that people thought of them as just for government, or big companies and, and in fact, they scared people. Now we've got it so they're fairly inexpensive, ah, individual people can own them, they're viewed as empowering, they let you reach out and learn things about the world. A big change will be that we'll be able to write software that models the world. So, for example, understanding if you're designing a product, you know, trying out, is this car going to be, ah, cheap? If it's in a crash, will it respond well? If I'm thinking of a medicine, ah, does it have the right shape, will it have the effect on people's bodies? So, being able to model the world will allow us to be way more productive.

Also in the time frame you're talking about, we'll even have robots, ah, computer controlled machines that can, ah, do hard tasks. Ah, today some robots, ah, clean up the carpet, but that's very simple – these robots will be able to see and walk, just like human beings. And so, ah, it'll, it'll be a radical change, ah, and that will probably happen even in 25 years.

Ah, for the next 25 years after that, you know, I think we'll all be surprised, because computing can go a long, long ways from where it is today.

Chris Bill, we have, ah, Olivia up here and Olivia has a question for you. Olivia, do you want to ask your question?

Olivia Um, do you think computers will ever be able to think for themselves?

Bill That's an excellent question and even within the field you find people who disagree. Ah, as we learn about human intelligence, our admiration for how powerful the brain is just goes up and up. The way that an infant can acquire vocabulary and common sense and a sense of time and, just pick up a book and learn new subjects, you know, we still have nothing that's even close to that.

In some areas like vision or listening, or even locomotion, moving around, the last five years there has been very good progress. And so, I think we can say for sure that we'll match human ability in terms of senses, ah, seeing,

smelling, tasting, being able to move, but when it comes to learning and the broad general purpose way that humans learn, ah, certainly nothing dramatic will change on that in the next decade. Beyond that, it's really a little bit guesswork. Ah, people like myself think yes, computers will eventually be smart. It's a little bit scary, ah, but getting the computer to be a little bit smarter and to be an even better tool, I think, can be a, a very positive thing.

Chris

Bill, perhaps we could finish with one last question, which is to do with an area that's very close to my own research interests which is, ah, use of software in medicine and health care. What do you think the exciting opportunities are there?

Bill

The most interesting area today, if I was a young person, is on that boundary between the very best computer science and medical advances. You know, I think about it a lot, not just in terms of the rich countries, but also the poor countries where you still have a lot of terrible diseases like malaria and AIDS and, you know, I know this is the golden age of using digital software, ah, to help improve the practice of medicine.

Chris

Bill, it's been a great pleasure talking to you and thank you very much for joining us.

Bill

Thank you.

Chris

So we've seen that there are plenty of important problems for computers to tackle. But after the break we'll be looking at problems which are so hard that even a supercomputer can be completely overwhelmed.

Part 3
Chris

Sometimes, even the most powerful computer in the world can be defeated by what looks like a simple problem. If I have a volunteer I can show you an example – who'd like to come on down?

Um, let's have you. Okay, come on. Just turn round there. What's your name?

Rhys

Ah, Rhys.

Chris

Rhys? Okay, Rhys.

What I have here is a very simple looking puzzle for you to solve. There are four pieces and all I want you to do is take these pieces and – that's it, you get a hold of it – and lock them together to make a two by two pattern there. Okay?

Now this looks like a fairly simple puzzle and you'd think that a computer would be very good at this, because after all, the computer just has to try lots and lots of different possibilities until it finds the answer. The problem comes if we make the puzzle a little bit bigger.

Here I have an example that's three by three, so this has nine pieces.

Now if I've got a computer that can look at a million different positions a second, then it would only take a blink of an eye to do that puzzle, but if I jumble up these pieces, it will take over a day to solve this.

Okay, how are we getting on? Shall I just give you a little bit of a clue here? There we go, okay, good – can you fit that one in? Okay, that's looking promising. Excellent, okay, well done, thank you very much.

So if my computer, doing a million moves a second, takes a day to solve that puzzle, what about this puzzle? This is five by five, so this has got 25 pieces in it. So if it takes a day to solve that one, how long do you think it would take to solve this one? Any idea? Four days? Any, any other ideas – a year? Somebody thinks it will take a year.

This is how long it would take to solve this puzzle if my computer tries a million moves a second – it would be 500 trillion, trillion years to solve this puzzle. That's a long time. The universe is a mere 14 billion years old.

Now of course we could speed it up a little bit, because the computer doesn't need to look at every possibility, but it would still take an astronomically long time to solve this puzzle.

You might think, well instead of trying lots of different possibilities, is there a really clever method that would allow a computer to solve a puzzle like this in, let's say, a few weeks? Well surprisingly, no one knows. It's one of the big open challenges of computer science.

And it's not just puzzles like this which are hard to solve. There are many practical problems, such as working out the perfect school timetable or, in my field, working out how to lay out the electrical components on a circuit board, that are so complex we simply have no hope of ever finding the exact solution.

But physicists and computer scientists are exploring a radically different kind of machine called a quantum computer which, one day, might be able to solve at least some of these very difficult problems. To see how that works, I've got a demonstration here.

In this box at the bottom, we have a source of microwaves and the microwaves are being beamed up here to this metal plate. Now this metal plate has two slits in it, so the microwaves are coming up through those slits.

And I have here a detector of microwaves and it's going to make a sound, and the louder the sound, the stronger the microwaves. Now you might expect that with two slits I would have two regions where the microwaves are strong and the sound is loud, but let's see what actually happens. Listen carefully to this.

Okay, it's getting stronger and quieter. Each of these yellow bands is a region where the microwaves are strong. So instead of having just two regions where the microwaves are strong we've got, ah, quite a few.

This seems very strange. What's going on here is the microwaves are coming up through those two slits and because they're waves, in certain

places the waves go up and down together and they add up and it's strong. But in between, the waves are going in opposite directions and they cancel out. So this tells us that the microwaves are coming through both slits at the same time.

Now if we repeat that experiment using light instead of microwaves and we use very small slits, we see this. Each of these little dots is a particle of light called a photon. But if we wait long enough and this, this whole experiment takes about half an hour to run, so we've, we've speeded it up here, the photons build up into vertical bands, or stripes, which are very much like the bands that we saw on the microwave experiment.

So that means that those photons, those particles of light, must have gone through both slits at the same time. Now this is an extremely peculiar quantum phenomenon, it's called superposition, the idea that something can be in two places at the same time. And it's not just photons that can be in two places at the same time.

This is a model of a molecule called Carbon 60, or a buckyball, each of these is a carbon atom. And molecules as big as this have been put into a superposition, in which the molecule is in two different places at the same time. Thank you.

But how can we make use of superposition to build a computer? Well to find out, we have two people who are wearing, I hope, very special T-shirts. Where, where are they, they're hiding somewhere? Yes, do you want to come on down?

Okay, if you'd like to come on down here. And what's your name?

Holly

Holly.

Chris

Holly, if you'd like to come and stand there. And your name?

Alex

Alex.

Chris

Alex. Now Holly and Alex are wearing very, very expensive T-shirts, because these are quantum T-shirts. Now, if you just stand back a second, let's imagine that Holly here is a, a bit, a binary digit in an ordinary computer. Now the computer could, first of all, process the value one, that would take one run of the computer. Now if you'd like to turn around, that's it, and stop there – good – the computer could then run a second time and process the value zero. So it would have to run two times.

Okay, you can turn round and face the front now.

But if you were a quantum bit, in a quantum computer, then you could be in a superposition of both 0 and 1 at the same time, and using the magic of television, we can see what that might look like.

So a quantum computer could process that superposition in one step, instead of in two steps like a classical computer.

So a quantum computer could be twice as fast as a classical, conventional computer. Now, that may not sound very exciting, but let's see what happens if we add in another quantum bit.

So if you'd like to come and stand there. And so, when you are both facing forward, you're both in the 1 state. If you turn around, both of you – that's good – now you're both a zero. And, if you turn back to the front again, they're actually – that's good, you can turn to face the front, that's good – there are actually four possibilities here, 11, 10, 01 and 10. So a conventional computer would have to process each of those four possibilities one after the other, it would have to run four times. But if you look up on the screen, we'll see what a quantum superposition would look like and that contains all four possibilities, so a quantum computer could do four things at once, so with two bits it would be four times as fast as a conventional computer.

And if you recall from earlier, every time we add another bit, we double the number of possibilities. So if we have a quantum computer with, let's say, 300 quantum bits, it would be able to do more things at the same time than

there are atoms in the universe and that's powerful enough to solve some of those tough problems that we talked about earlier.

Now quantum computers won't solve all of those problems, and they won't be general purpose computers, but one day they could solve problems that simply can't be solved in any other way. Okay, you can go back to your seats now, thank you very much.

Now, you might have noticed that you can't buy quantum computers in the shops just yet and there's a, there's a good reason for that. The reason is that this superposition state is incredibly fragile, even the tiniest disturbance will destroy it. At the moment, the most complex calculation that's ever been done by a quantum computer, is to show that the number 15 is equal to 3 times 5, so not very impressive. But, there are lots and lots of ideas for ways to build practical quantum computers so, in perhaps 30 year's time, when some of you are scientists, we might see the first large scale quantum computers.

Until then, we're faced with computational problems that are so difficult there is simply no hope of any computer solving them. Now that might sound like bad news, but in fact, we can use it to our advantage. Every time we send secret information over the internet, such as our name and address or credit card information, we're relying on these very hard computational problems in order to keep it secure. To find out why, join me for tomorrow's lecture, when we'll be untangling some of the secrets of the web. Thank you.