Where to next, Joe? P versus NP, explained.

by Jack Townsend

Meet Joe. He and his family—two kids, a dog, and his wife Karen—are going on a road trip. They're trying to plan their route and book lodging. They've got a few dozen places to visit, but they don't care about the order of their stops. They do, however, want to find the absolute shortest way around the country that passes through each of the places they'd like to visit.

Now say you're a computer scientist and Joe is asking you to help him figure out which route you should take. You start thinking about the problem, but you can't figure out many tricks. So you write a program that takes each conceivable route, calculates its length, and then keeps track of the shortest option out of all the routes the program has considered so far. One evening, as you're leaving the office, you start the program. When you come back in the next day, the program is still running, but it's hardly made any progress.

You break out a calculator and find that the program won't complete for a while. A long while. A while so long that it's longer than the age of the universe. So you tell Joe and his family that they'll just have to guess the shortest possible route. You come up with some ways to make their guesses more accurate, but in the back of your mind the real problem—finding the absolute shortest route—lingers.

This problem, finding the best route for Joe's road trip, is a member of the group of problems called the *NP-hard* problems. But if you change the problem a little bit so that you're in search of any route whose total length is less than a particular number of miles (what number exactly doesn't matter) then the problem is still NP-hard, but it is also *NP-complete*. Answers to this version of the problem would be easy to verify—you would just plug it in to Google Maps and see how many miles the route is—but finding that correct solution would still be fantastically hard. The problem would get harder, and would eventually be impossible, as you ask for shorter and shorter routes. This is because there are fewer short routes than long routes. The technical details aren't very important here. What matters is that we don't know how to solve the problem.

It turns out that Joe's problem is a member of a group of problems that are all clustered around a single mathematical principle. We don't understand that principle, but we do know that if someone discovers it, they will have solved all of the other problems in the cluster. Most of the problems, these NP-complete problems, are very abstract. But they can be applied to real-world problems like Joe's roadtrip. (The abstract version of Joe's problem is called the Travelling Salesman Problem.) Depending on the circumstances, creating new vaccines is sometimes another one of these problems. As it stands, finding vaccines for new diseases is a radically hard problem. But Joe's problem and the vaccine problem are one in the same: solve one, and you've solved the other. The applications extend far beyond road trips and even medicine. For instance, creating a masterpiece is a hard problem, but looking at art is an easy one. Hitting a home run? Hard. Watching the batter swing the bat? Easy. A solution to Joe's problem might mean that the hard problem in each of those pairs would become as easy as the easy half of the pair.

By watching the ball sail over the yellow line at the top of the left field fence, you can verify that a hit is in fact a home run. Verifying the solution—watching the ball—is currently much easier than creating it—that is, hitting the home run. Abstractly, a solution to Joe's problem would render the hard problem of home run hitting easy. Mathematicians ask whether or not this is true in general—whether or not the solutions to these difficult problems are easy to find. They call this question the P versus NP problem, where P means "polynomial time," and NP means "non-deterministic polynomial time." The acronyms refer to two types of Turing machines, which are theoretical models of computers named after famed WW2-era British computer scientist Alan Turing. The problem can be solved in two ways. First, P could in fact be equal to NP. This is true if Joe's problem can be solved quickly and hitting home runs and watching baseball games are of commensurate difficulty. This result is expressed as $P = NP$. Second, P could be unequal to NP, or $P \neq NP$. This second result is the one most experts believe is correct, and it jibes with our usual understanding of the world.

It's obvious to us that some things are harder than others: creating art is harder than looking at it, hitting home runs is harder than watching a baseball game. So the million-dollar question (literally—the Clay Foundation has offered $1 million to the first person to solve this question) is: can certain problems which we think are hard actually be solved easily? Is solving hard problems quickly just a matter of finding better solutions, or are certain problems inexorably hard? But this question is even bigger than creating masterpieces with a computer program.

This problem captures an essential challenge of the human condition. Difficulty and limitation are key components of life. We cannot fly, and we cannot know the true nature of God, if one exists. But we can make our own decisions and do virtually anything we put our minds to. We take these dual realities for granted—perhaps they seem inescapable. Work—a third of our waking lives, in the typical American formulation—stems from our conviction that value does not simply exist but is created through creativity and time and effort. As a society, we are caught between two realities: that we cannot do everything, and yet that we can do anything. In short, humanity is blessed with creativity and cursed by fallibility.

And math is supposed to be the way out of our existential, permanent tension. Math is supposed to be the purest expression of the universe's essential truths. But very often, someone will get a quick laugh by making fun of how bad at math they are; many, many people think that math is an impenetrable, even useless, abstraction. Yet, since the 1950s, one of the most important questions in math has been about the uniquely human relationship between the truth and our knowledge of the truth. As it turns out, Joe's problem represents fundamental questions of both human nature and math.

By now, it's clear that P versus NP is much more than just a math problem. It's another way to talk about the human condition, and solving it will either affirm our intuition that humanity is simultaneously limited and yet infinitely creative, or it will upend human endeavor as we know it. Depending on the answer to this problem, art and a thousand other things humans spend time doing will either continue to be mysterious or become just another algorithm that computers do better than humans.

Despite most experts' opinion that P $\neq$ NP, the road to solving the unsolvable questions does not end here. Quantum computing, a subject so esoteric that no one seems to know anything beyond its name, may hold the keys to answering the questions that aren't quickly solved with traditional math and computer science.

Math expresses the simplest truths. If P $\neq$ NP, it will also express our most profound limitations. But each time we learn another of our limitations, we are only taught another way not to solve our problems. We get closer to the answers.

Math has a way of evolving to meet new challenges. Really, it may be human nature. We take on new challenges not because, strictly speaking, it's necessary that we do so. We take on the impossible for the same reason JFK stood before the nation and said, "this generation does not intend to founder in the backwash of the coming age of space."

Nor do we, in these initial years of the digital age, intend to linger for long where we are. The human spirit is alive in P versus NP and we do not intend to hesitate before the next leap forward.