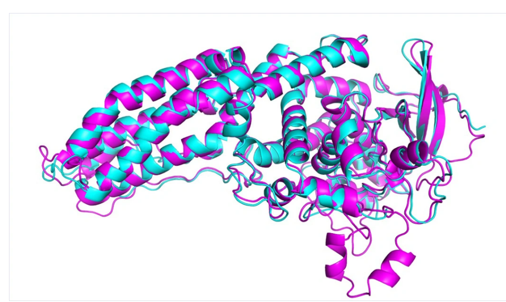
Could curing cancer truly be as easy as 2 + 2? P vs NP - one of the biggest problems in modern computing and mathematics .

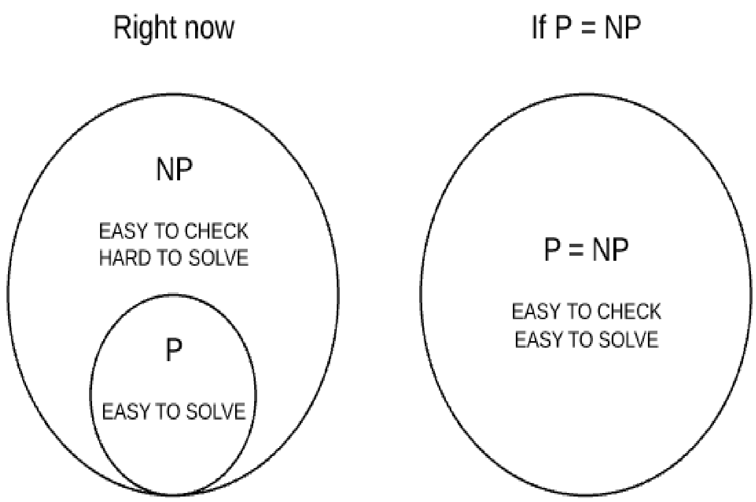
Denim Adewumi

Could finding the cure to cancer be ultimately reduced to a trivial task? How would you like to be able to beat Candy Crush as easily as you could list the numbers from one to ten in size? For many I'd imagine the answer to such questions would be a definite yes. Well as far as Mathematicians and Computer scientists alike are currently aware, this very well might be possible, but hold onto your horses. The ultimate question surrounding these two hypotheticals goes so much deeper than would meet the unknowing eye and really highlights some of the beauty and important wider applications of the abstract concepts in mathematics that we may sometimes mistakenly call purely theoretical.

Suppose I asked you to factorize a number such as ten. This wouldn't really pose much of an issue for a person, nor a computer to solve. You can determine without much thought that its factors would be five and two. You can also very easily verify, if given the answer, that five and two multiplied together do in fact combine to make ten. Now consider if I asked you to do the same to a number that contained a million digits within it. This would actually be a whole lot harder for not only a human to solve but even the most powerful supercomputers as well. By which I mean the time taken would very much be in the realms of longer than the age of the universe to compute. That is because the rate of difficulty, that is time to compute, goes up at an exponential rate to the size of the number of items. However if I gave a computer the solutions to the problem, say two numbers, the computer could very easily test if they do in fact multiply to make the original number. That's pretty cool but what does that have to do with curing cancer or beating Candy Crush? Well you see this is what is known as non deterministic polynomial problem, or NP for short, and as it happens those two other situations have been shown to also be NP problems. What this means is that they can all, on a fundamental level, be broken down into the exact same problem. With cancer, the problem of protein folding is what was shown to be NP and if solved would allow for a cure to cancer to be found. Now before you get confused on all of the mathematical jargon and terminology, what a non deterministic polynomial problem is, in layman's terms, is a problem that is typically hard to solve but easy to verify (although having said that, just because a problem is NP does not mean that it's inherently or for all intents and purposes impossible to solve). An alternative to such a problem would be one such as sorting the numbers in a list. This is one that would typically be easy to solve and easy

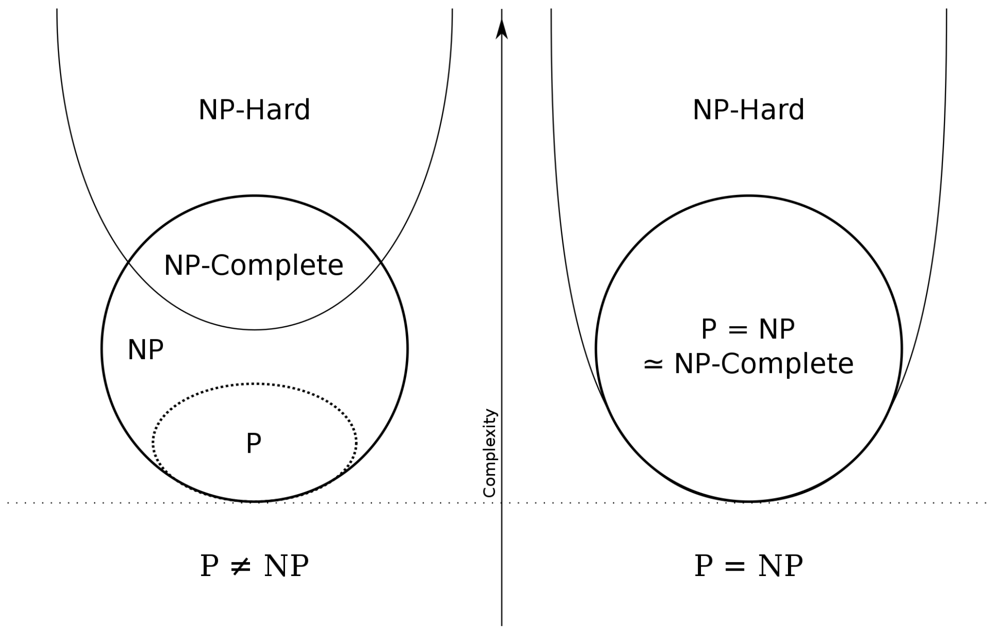


to verify (but again not always and this is talking about for a computer) and this is what is known as a polynomial or P problem. It gets this name from the fact that the time taken to solve goes up at a polynomial rate with the number of items which is of course independent of absolute difficulty hence my emphasis on the word “typically”. Then polynomial just means that it goes up at a rate of x^n as opposed to n^x which would be the case with an NP problem where x is variable and n is a fixed positive integer. For example, one of the most common sorting algorithms used by computers known as bubble sort has a big O of x^2 and therefore would make it a P or polynomial problem (big O notation is a whole ‘nother discussion for another time but that’s again just referring to the rate the run time increases with the number of items). The question that mathematicians and computer scientists would like to answer is whether or not all NP problems could be ultimately reduced to a P problem or simply does P=NP which would essentially mean that the set of all NP problems is contained within the set of all P problems. The problem was initially stated and introduced in 1971 by Stephen Cooke and then independently by Leonid Levin in 1973.



It’s worth noting that most experts in the field do not in fact believe that P is equal to NP. That is to say that the set of all NP problems can not be ultimately reduced into the set of P problems. With that being said, none have yet to prove it either way despite this being a known problem for more than fifty years, which should tell you that this is certainly no trivial task and would be a revolutionary proof once complete. There have been countless attempts but those were all unfortunately in vain and these days only proofs that consider the problem in a unique approach would even begin to be considered.

Just because the problem has not yet been solved however does not mean there have not been any advances made towards doing so in the last few decades. To begin with one, experts within the field have come up with a concept known as NP completeness which is very useful in tackling the problem of P vs NP. NP-complete problems are a set of problems to each of which any other NP-problem can be reduced in polynomial time and whose solution may still be verified in polynomial time. That is, any NP problem can be transformed into any of the NP-complete problems. Informally, an NP-complete problem is an NP problem that is at least as "tough" as any other problem in NP. Furthermore, NP-hard problems are problems in which they are at least as hard as NP problems which essentially just means all NP problems can be reduced (in polynomial time) to them. NP hard problems don't actually have to be contained within NP. Now by the very logic presented it should follow that if any NP-complete problem is contained within P then P=NP. There are many many important problems that have been shown to be NP complete however there is unfortunately not a fast algorithm for them. As it happens the first problem that was proven to be NP-complete was the boolean satisfiability problem and is also known as SAT and is given directly by the Cook-Levin theorem. Following this, a method known as proof by reduction was used to more easily show that many other problems are NP-complete including sudoku. In this case, the proof shows that a solution of Sudoku in polynomial time could also be used to complete Latin squares in polynomial time. This in turn gives solutions to many other problems which could in turn be used to solve other NP problems in polynomial time. These concepts can be difficult to follow whilst reading but can be summarized in the diagram given below which is essentially an extension to the one previously shown.



You might think that if P were to be equal to NP then the world would be a much better place than it is currently, and whilst it's true that the world would be a much different place, there could also be negative consequences in such an event. For example, most modern security and encryption systems rely on the notion that P is not equal to NP. If this was found to be a false notion then all the world's passwords and bank information could be rendered trivial to figure out and solve which is just one example considering the flip side of the argument, there are many others. This was just a basic introduction into one of the most important topics in mathematics and the only true way to do the topic any justice would be to go and study it past degree level but for the purpose of this article, my explanations should have hopefully been more than sufficient. The most interesting point however is perhaps a mathematician could be the one responsible for what leads to a cure for cancer and so many other life altering changes. I suppose this however is still something that remains to be seen.

References
Information from this article primarily came from the wikipedia article on P vs NP and various other random sources with the information discussing NP-Completeness coming directly from wikipedia with some changes. The first image is a clip from the tv show Futurama showing an easter egg in which P vs NP is depicted. The image of protein folding came from an article written by Will Douglas Heaven. Finally the images with the euler diagrams came from an article discussing P vs NP written by Bilal Aamir and Wikipedia respectively.

https://en.wikipedia.org/wiki/P_versus_NP_problem
https://medium.com/@bilalaamir/p-vs-np-problem-in-a-nutshell-dbf08133bec5
https://www.technologyreview.com/2020/11/30/1012712/deepmind-protein-folding-ai-solved-biology-science-drugs-disease/