Zachary Suzuki and Kevin Leong

Dr. David Brauchler

CS 2353

27 April 2021

Does P = NP?

To start this question off, we must first define what P is, and what NP is. P is defined as the general class of questions for which there exists an algorithm that can find an answer in polynomial time. NP is defined as the general class of questions for which there exists an algorithm that can verify that an answer is correct in polynomial time. The statement, therefore, says that there exists a general class of questions for which the solution and the verification of that solution can be found in polynomial time. It is reasonable to assume that P is a subset of NP, because for any problem that can be solved in polynomial time P, there exists a way to verify that solution in the same time (polynomial time) or less. This is easy to see intuitively, as you could just resolve the problem in that same amount of time, and that would verify the solution.

One concept that was introduced as a way to perhaps prove or disprove this statement is the idea of NP-completeness. NP-complete problems are the set of problems that any NP problem can be reduced to. What this means is that if an NP-complete problem has an algorithm that can find its solution polynomially, this would imply that P = NP, since one NP-complete problem can represent all other NP problems theoretically. So far however, there is no algorithm written to date that can solve any of the NP-complete problems polynomially. Thus, for all intents and purposes, P does not equal NP.

Aside from this absence of a proof in the positive, there are many other reasons why it is unlikely that P = NP. First, it makes sense that it is quicker to verify a solution than it is to solve

the problem associated with it. Second, the theoretical polynomial-time algorithm to solve a problem only works if it is already assumed that P = NP. Additionally, the use of this algorithm would require unreasonably large constants, which make the algorithm impractical. Third, the existence of P = NP would disprove many other falsities in the math space. Ultimately, this problem is not solved, but there exist many things that function only because its nonexistence is assumed.

References

https://en.wikipedia.org/wiki/P_versus_NP_problem

https://www.britannica.com/science/NP-complete-problem

https://techterms.com/definition/cryptography

https://en.wikipedia.org/wiki/Natural_proof