Advances in Theoretical and Applied Mathematics.
ISSN0973-4554 Volume 19, Number 2 (2024), pp. 37-40

http://www.ripublication.com/atam.htm

Proof of the P versus NP Problem

Daniel John Thompson

La Trobe University
133McKoy St West Wodonga Vic, 3690, Australia

Abstract

Within this paper I show that P ≠ NP. Obtaining two exponential functions from two logarithmic functions demonstrates that both the complexity and the polynomial time it takes to solve the two different equations are different. It can be seen that for problems in P, the complexity and the time it takes to solve the problems does not change for all problems in this class. However, it is clearly shown that for problems in NP the increase in complexity causes an increase in the polynomial time it takes to solve such problems. The polynomial time it takes to solve NP problems increases exponentially as the complexity increases. This proves that deterministic problems in P are not equal to non-deterministic problems in NP. Despite the verification for NP problems being decided quickly once solved, this is not enough to place P problems in the same class as NP problems.

Keywords: P vs NP problem, computability theory.

Introduction

The P vs NP theory is based on the successful theories of NP – completeness and complexity-based cryptography. NP - completeness theory originates from the work of Turing, Church and Gödel in the 1930s and stems from computability theory. The computability precursors of the classes P and NP are the classes of deterministic and non-deterministic languages, respectively. The P versus NP problem determines if a nondeterministic algorithm in polynomial time is also accepted by some (deterministic) algorithm in polynomial time (Cook, 2019, p.1). A P problem is one that can be solved in polynomial time and has an algorithm for its solution (Hosch 2019, p.1). An NP problem is one that can be guessed with no rule followed to make the guess and can be verified in polynomial time (Hosch 2019, p.1).

Problem statement: Does P = NP? (Cook, 2019, p.2).

Method

By using a system of two exponential functions derived from two logarithmic functions a solution to the P versus NP is found.

$x = \log_b y$ then $y = b^x$ (Washington 2005,p.373).

Let b equal the complexity of the problem in both P and NP.

Let P equal polynomial time problems and NP equal non polynomial time problems.

Considering problems in NP are more complex to solve they would be considered exponentially harder to solve than those in P.

Let x equal the time it takes to solve a problem for both P and NP problems.

These statements lead to the following equations relating to problem in P and NP, the time it takes to solve them and the time it takes to verify these problems.

Let x equal the following.

$\log_b(\log P) = x$ then $b^x = \log P$ to solve NP problems let $b^{\log(\log P)} = NP$

$$\therefore \quad b^x = \log P = NP \Rightarrow b^{NP} = P \quad \textbf{(1)}$$

$\log_b(\log NP) = x$ then $b^x = \log NP$ to solve P problems let $b^{\log(\log NP)} = P$

$$\therefore \quad b^x = \log NP = P \Rightarrow b^P = NP \quad \textbf{(2)}$$

Firstly, to solve and verify problems in P let the complexity b equal to 1 and NP as the exponent representing the time it takes to solve that problem.

From equation (1) solving for P problems in polynomial time.

$b^x = \log P = NP$

$b^{NP} = P$

$1^{NP} = 1$ for all integers representing NP.

Secondly to solve problems in NP let the complexity b, be greater than 1 as it is more complex and the polynomial time P it takes to solve the problem also greater than 1.

From equation (2) solving for NP problems in non-polynomial time.

$b^x = \log NP$

$b^P = NP$

b^P = is even as b is even and as for all even integers n, $n^{odd \vee even}$ is even (Stanford.edu, 2024: Farlow, 2020).

As complexity b increases.

$b+1^P$ = is odd as b is odd and as for all odd integers n, $n^{odd \vee even}$ is odd (Stanford.edu, 2024: Farlow, 2020).

When the complexity is increased the polynomial time, it takes to solve the problem in NP is increased exponentially.

$P \neq NP$

QED.

Results

When a P problem is solved, its complexity is easy, and it is easy and quick to verify as seen from equation (1). Due to the simple complexity of the P problem, the time it takes to solve and verify the solution is the same for all problems in P regardless of the time represented in the equation. When a NP problem is solved, as the complexity increases and the polynomial time it takes to solve the problem increases the solution is exponentially greater than problems in P.

Conclusion

From the results it is concluded that $P \neq NP$. The polynomial time it takes to solve problems in NP is exponentially increased as the complexity of the problem is increased compared to problems in P. Problems in P are easy to solve and the polynomial time it takes to solve the problem is equivocally the same for all problems in P.

References

[1] Cook, S 2019, Clay Mathematics Institute, *P versus NP problem*, 2019

viewed 21st June 2019 https://www.claymath.org/millennium-problems/pvs- np-problem

[2] Hosch, WL 2019, *P versus NP problem,* viewed 21st June 2019,

https://www.britannica.com/science/P-versus-NP-problem

[3] Farlow, SJ, 2020, *Advanced Mathematics*, John Wiley & Sons, Inc. Hoboken, NJ 07030 USA.

[4] Stanford.edu, (2024). *Mathematical Proofs, Stanford.edu*

https://web.stanford.edu/class/archive/cs/cs103/cs103.1202/lectures/01/Small01.pdf

[5] Washington, AJ, 2005, *Basic Technical Mathematics with calculus*, Pearsoneducation Canada Inc., Toronto Ontario.