P Vs NP: The '70s Bodyguard of Modern Encryption

Humans are naturally secretive and it has been seen time and time again which is seen from the Ancient Egyptians to the front line battlefields of today's society. Cryptography stems from the Greek to mean "Secret Writing" and takes many different forms but is intrinsically related to mathematics. Our modern ciphers rely on the use of very large prime numbers that employ the P vs NP problem to postulate that with modern technology these are practically impossible to solve and so everyone can use them to safeguard data to be exchanged. The P vs. NP problem[1] is that for every problem that has an efficiently verifiable solution we can also find the solution efficiently which is one of my favourite mathematical ideas.

The implications of P vs NP on the fields of maths, computer science and many multi-disciplinary industries were recognised in the Clay Mathematics Institute's Millennium problems which are seven puzzling questions that have persisted across the history of mathematics and have not been solved, with each problem having a $1 million prize fund.[2] The P vs NP problem was formally coined by Stephen Cook and Leonid Levin independently in the early 1970s. The millennium prize fund for P vs NP is based on the conjecture that there is no feasible way to solve the easily verifiable problems using the assistance of computation which no one has been able to prove rigorously.

The notion of P or P TIME is used to demonstrate problems that can be solved using a Deterministic Turing machine in Polynomial Time and NP is Non-Deterministic Polynomial Time and so can be

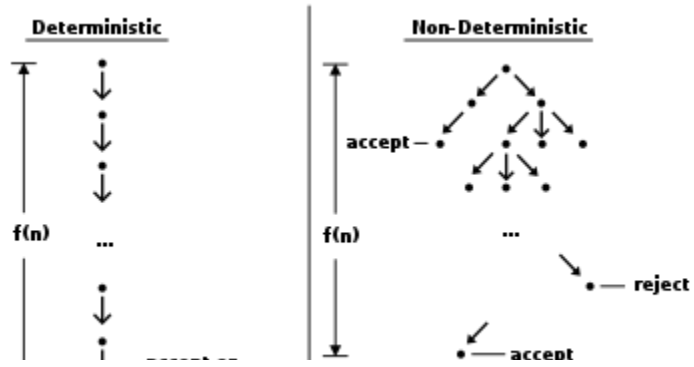

Figure 1 Deterministic Vs. Non-Deterministic Turing Architecture

[1] Fortnow, L., 2009. The status of the P versus NP problem. Communications of the ACM, 52(9), pp.78-86. accessed 20 March 2021 (http://people.cs.uchicago.edu/~fortnow/papers/pnp-cacm.pdf)
[2] Clay Mathematics Institute, P vs NP Problem, accessed 20 March 2021 (https://www.claymath.org/millennium-problems/p-vs-np-problem)

verified easily but not solved in Polynomial Time by a Deterministic Machine. Deterministic Machines are Turing Machines (these simulate computation) which we see almost every day and execute one action per state of the machine comparatively, they differ from Non-Deterministic Machines as they can do more than one action per state. Polynomial Time is also fascinating as it is simply put an expression of time using a polynomial equation, but the NP problems cannot be solved on the scale of Polynomial Time but can be verified in Polynomial Time on a Deterministic Machine but could be solved in polynomial time on a Non-Deterministic Machine which highlights their theoretical importance.

Modern cryptography revolves around the idea that each person has a private key and a public key[3]. The public key is made available to everyone and if Alice wants to send a message to Bob then she uses Bob's public key to encrypt it. Only Bob then has the private key who can decipher the message that Alice has sent. One such application of P vs. NP is shown below in RSA Encryption:

THE RIVEST-SHAMIR-ADLEMAN PUBLIC KEY SCHEME

Design

Find two large prime numbers p and q, each about 100 decimal digits long. Let $n = pq$ and $\psi = (p - 1)(q - 1)$.

Choose a random integer E between 3 and ψ that has no common factors with ψ. Then it is easy to find an integer D that is the "inverse" of E modulo ψ, that is, $D \cdot E$ differs from 1 by a multiple of ψ.

The public information consists of E and n. All other quantities here are kept secret.

Encryption

Given a plain text message P that is an integer between 0 and $n - 1$ and the public encryption number E, form the ciphertext integer

$C = P^E \bmod n$.

In other words, raise P to the power E, divide the result by n, and let C be the remainder. (A practical way to do this computation is given in the text of Hellman's paper.)

Decryption

Using the secret decryption number D, find the plain text P by

$P = C^D \bmod n$.

Cryptanalysis

In order to determine the secret decryption key D, the cryptanalyst must factor the 200 or so digit number n. This task would take a million years with the best algorithm known today, assuming a 1μs instruction time.

Figure 2 RSA Algorithm

This is the most used method that is currently being used on the billions of devices connected to the internet and allows us to freely share information that is sensitive without it being compromised is called the RSA algorithm and is detailed above. The mathematical beauty of this system is extraordinary as pairing the modulus function paired with 200 digit long semi-prime numbers we can

[3] Hellman, M.E., 2002. An overview of public-key cryptography. IEEE Communications Magazine, 40(5), pp.42-49. accessed 20 March 2021 (https://netlab.ulusofona.pt/im/teoricas/OverviewPublicKeyCryptography.pdf)

create an NP problem that would take over 1 million years to decode with current technology. In the RSA algorithm we can only decode a message if we find the p and q values by factorising n, but the computational time to solve the problem, even when computers can do $3.9x10^{18}$ operations per second[4], is an unfeasible amount of time which is how mathematics safeguards our latest Amazon orders. If P=NP, which describes the situation where an easily verifiable algorithm is also easy to solve in case studies such as the RSA, this means that the security of the current world would be fractured. Cryptography has been the foundation of governments and individual communication for millennia and must evolve alongside our technology, a polyalphabetic substitution cipher device from the 1500s is shown below which was advanced for the time period.

Figure 3 Ancient Ciphers predating the modern RSA system

Focusing on the NP side of the argument, there is a spectrum in classifying the decision problems. The cryptographic issue of factoring primes that I discussed earlier has resisted being branded as NP-Hard and is likely to be NP-Intermediate which could suggest that there is some algorithmic approach that does not encroach on the P=NP question.[5] NP-Hard problems are a class of the NP group that are the most difficult to solve in the NP class and with this falls out other terms such as NP-Intermediate

[4] Dongarra, J., 2020. Report on the Fujitsu Fugaku system. University of Tennessee-Knoxville Innovative Computing Laboratory, Tech. Rep. ICLUT-20-06. accessed 21 March 2021 (http://performance.netlib.org/utk/people/JackDongarra/PAPERS/icl-utk-1379-2020.pdf)

[5] Papadimitriou, C.H., 1997, July. NP-completeness: A retrospective. In International Colloquium on Automata, Languages, and Programming (pp. 2-6). Springer, Berlin, Heidelberg. accessed 21 March 2021 (https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.32.8685&rep=rep1&type=pdf)

and NP-Complete. The NP-Complete subset is particularly interesting as it is defined by the fact that all problems in NP can be morphed into the NP-Complete problem in a Polynomial-Time which is quite technical but results in NP-Hard also agreeing with this condition but they do not have to be easily verifiable (not in NP itself). As NP problems are said to have at most an exponential run time some operations have a super-exponential run time and so have been proven to be unsolvable in the P Vs NP context. One such example is Presburger Arithmetic where a computational algorithm can be used to decide whether an input is defined in Presburger Arithmetic from its laws or axioms relating to the addition of binary numbers. Even though this is a seemingly infinite time to compute at the worst case, this does not prove P ≠ NP as only in a certain condition is this problem accepted into NP in which case it still has an exponential run time only and so the super-exponential runtime only exists outside of the NP class and the P vs NP problem.[6]

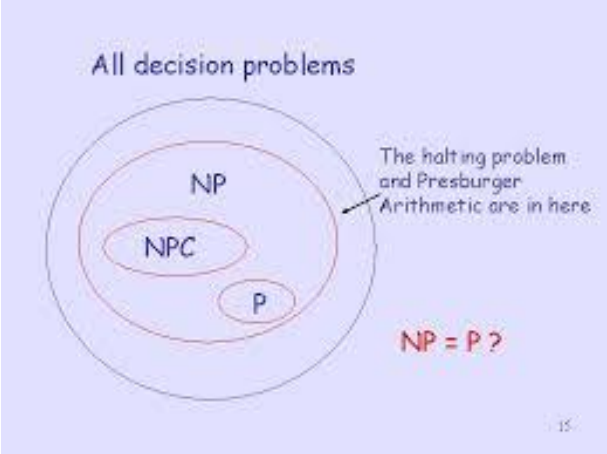

Figure 4 Depiction of the NP class and relative positions of NP-Complete, P and Presburger Arithmetic

In an age of advancing technology, one thing seems to come close to the P vs NP problem which is the theory of quantum computation. From our previous definitions, it seems that quantum computation would be Non-Deterministic and so could solve the NP problems on a polynomial time scale but there is a subtle difference in that a Quantum Turing Machine (QTM) has several inputs in a superposition that can all happen at the same time and so makes it far more powerful when combating

[6] Fischer, M.J. and Rabin, M.O., 1998. Super-exponential complexity of Presburger arithmetic. In Quantifier Elimination and Cylindrical Algebraic Decomposition (pp. 122-135). Springer, Vienna. accessed 22 March 2021 ((https://link.springer.com/chapter/10.1007/978-3-7091-9459-1_5)

P vs NP problems. However, the conjecture that QTM's are far superior to the current quantum technologies has not been proven yet [7]. The Non-Deterministic Turing Machine on the other hand has an output that does not depend on the input and tends towards beneficial decisions among an exponential number of options. This means that quantum machines that operate at millions of times the speed of current computers can use algorithms such as the Shor algorithm to factor primes in a polynomial time scale. This specific algorithm is one of the biggest breakthroughs in quantum computation but as mentioned before, the factoring problem is not NP-complete and so once again does not stand on the same level of the P vs NP battleground.[8] When quantum computers become commonplace our current RSA system will crack under this computational pressure but we can use an approach such as the "BB-84 Protocol" to utilise the quantum technologies to maintain security but this digresses from the P vs NP problem yet will be integral to the privacy of our future.

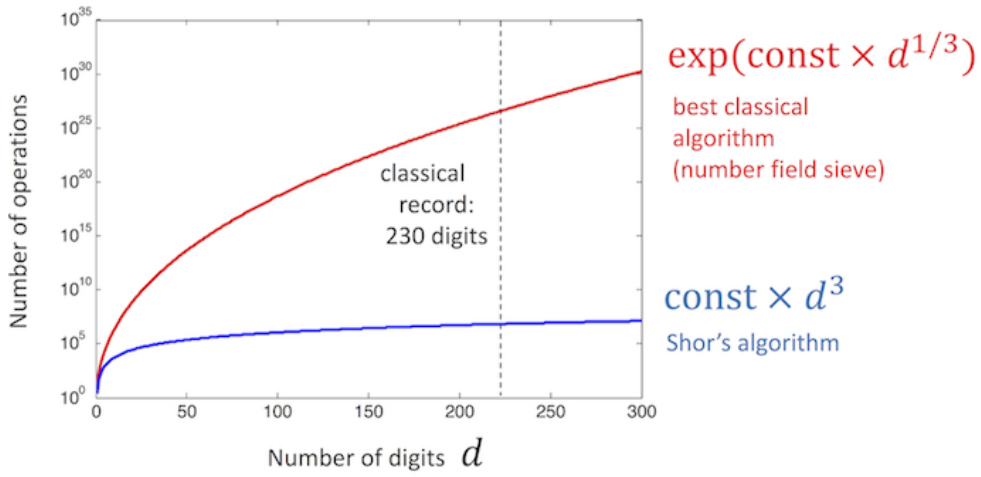

Figure 5 Shor Vs Traditional algorithm for factoring primes

Overall, the P vs NP problem is fundamental to many aspects of our lives and I discovered the conjecture when looking into how we encrypt our data. The beauty of the problem is that it has been unsolvable in the field of Maths and Computer Science for many years and even though it has a very significant impact on many disciplines it remains a mystery. There is still much more to learn but over

[7] Tusarova, T., 2004. Quantum complexity classes. arXiv preprint cs/0409051. accessed 22 March 2021 ((https://arxiv.org/abs/cs/0409051)

[8] Preskill, J., 1998. Quantum computing: pro and con. Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences, 454(1969), pp.469-486. accessed 22 March 2021 (https://arxiv.org/pdf/quant-ph/9705032)

with over 1000 papers on the topic I look forward to delving deeper and perhaps contributing in some way in the future.

Reference List:

Clay Mathematics Institute, *P vs NP Problem*, accessed 20 March 2021 (https://www.claymath.org/millennium-problems/p-vs-np-problem)

Dongarra, J., 2020. *Report on the Fujitsu Fugaku system*. University of Tennessee-Knoxville Innovative Computing Laboratory, Tech. Rep. ICLUT-20-06. accessed 21 March 2021 (http://performance.netlib.org/utk/people/JackDongarra/PAPERS/icl-utk-1379-2020.pdf)

Fischer, M.J. and Rabin, M.O., 1998. *Super-exponential complexity of Presburger arithmetic. In Quantifier Elimination and Cylindrical Algebraic Decomposition* (pp. 122-135). *Springer, Vienna*. accessed 22 March 2021 (https://link.springer.com/chapter/10.1007/978-3-7091-9459-1_5)

Fortnow, L., 2009. *The status of the P versus NP problem*. *Communications of the ACM*, 52(9), pp.78-86. accessed 20 March 2021 (http://people.cs.uchicago.edu/~fortnow/papers/pnp-cacm.pdf)

Hellman, M.E., 2002. *An overview of public-key cryptography*. *IEEE Communications Magazine*, 40(5), pp.42-49. accessed 20 March 2021 (https://netlab.ulusofona.pt/im/teoricas/OverviewPublicKeyCryptography.pdf)

Papadimitriou, C.H., 1997, July. *NP-completeness: A retrospective. In International Colloquium on Automata, Languages, and Programming* (pp. 2-6). *Springer, Berlin, Heidelberg*. accessed 21 March 2021 (https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.32.8685&rep=rep1&type=pdf)

Preskill, J., 1998. *Quantum computing: pro and con. Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 454(1969), pp.469-486. accessed 22 March 2021 (https://arxiv.org/pdf/quant-ph/9705032)

Tusarova, T., 2004. *Quantum complexity classes*. *arXiv preprint cs*/0409051. accessed 22 March 2021 (https://arxiv.org/abs/cs/0409051)