

REVIEW OF *THE GOLDEN TICKET*

This short book gives a decent explanation of the P vs NP problem and its significance in nontechnical terms. My understanding of it had been pretty hazy until taking a course this semester that required writing some NP-Completeness proofs, and apparently I still have a poor grasp of the extent of NP, because some of the things Fortnow says would be implied by $P=NP$ were surprising to me:

Suppose we could simply describe a task and immediately have a program that provided that functionality. Feed in a movie of a human tying a knot, and immediately have the computer repeat the process with robotic hands. Give a computer the complete works of Shakespeare, and have the computer create a new “Shakespeare” play. Suppose whatever we can recognize we can find. We can if $P = NP$.¹

I’d have liked more information on how these problems would be formulated such that a solution can be verified in polynomial time. I guess the first one might go something like: construct a sequence of actions which, when run in a physics simulator and rendered from a particular angle, produces a video matching the input. (Relatedly, there’s an example on page 17 about being able to immediately render a baseball game from any angle using video from several cameras.)

I’m less clear on the second one. Maybe you could view it as searching for an ML model that produces output that scores highly according to a classifier model trained on Shakespeare’s plays? ChatGPT is quick to assert that “the problem of verifying the ‘Shakespeare-ness’ of a play is in NP”—let’s take a moment to delight in its completely spontaneous use of the term “Shakespeare-ness”—but when pressed for details, it lists criteria like “linguistic features”, “thematic elements”, “character complexity”, and “plot structures” which sound hazy. I’m curious exactly what Fortnow had in mind.

Anyway, the book talks through some standard examples of NP-Complete problems like clique and vertex cover (using cutesy framing for the math-averse) and mentions some I hadn’t heard of before like “[f]inding the minimum energy state of a physical system such as interactive magnetic particles or soap bubble formation”². It was also interesting to hear that Minesweeper and Tetris are both NP-Complete.

¹Lance Fortnow, *The golden ticket* (Princeton University Press, 2017), 6.

²Ibid., 48.

I appreciated the book's discussion of history, covering both the Western and Russian development of the question and giving a very brief overview of some of the approaches for proving $P \neq NP$ that have been tried so far.

Before reading the cryptography chapter, I didn't know that it's possible to provide zero-knowledge proofs (i.e. evidence you can send to someone, which they can verify for themselves, to prove that there's a solution to a particular instance of the problem, without giving them any details of the solution) for all NP-Complete problems. That's pretty cool!

There's a brief mention of "NC", the class of "problems that computers can solve quickly in parallel"³. This statement was surprising to me so I think it'd be interesting to read more about:

...we don't even know if $NP = NC$. $NP = NC$ means every NP search problem can be solved extremely quickly on systems with many computers and/or cores, an even more beautiful world than that of $P = NP$.⁴

³Ibid., 157.

⁴Ibid., 157–58.