Is P=NP?

Task - 5

Made by -
Shivam Joshi

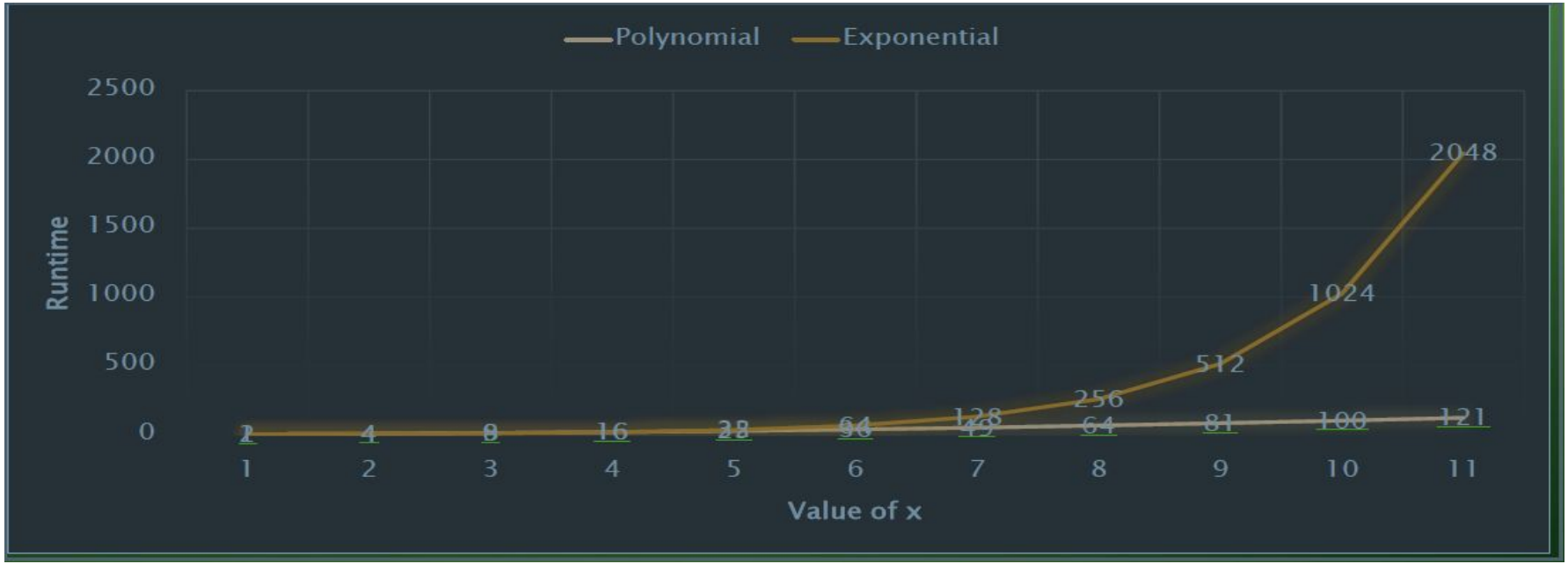
Measurement of better algorithm

How long does it take to execute a given algorithm? In CS the answer is not given in minutes/sec. but relative to the number of elements the algorithm has to manipulate.

Polynomial vs Exponential

Polynomial n^x

Exponential 2^x



Proof that the given classes take the same time or not and you can get 1 million dollars

1. **P class** is a set of problems whose solutions running times depend polynomially on the size of the input. Thus, it is relatively easy to find these solutions using programs that are reasonably fast.
2. **NP class** is a set of problems whose solution is very hard to find – perhaps requiring billions of years worth of computation – but once found, it is easily checked in polynomial time.

Example

Consider Sudoku, a game where the player is given a partially filled-in grid of numbers and attempts to complete the grid following certain rules. Given an incomplete Sudoku grid, of any size, is there at least one legal solution? Any proposed solution is easily verified, and the time to check a solution grows slowly (polynomially) as the grid gets bigger. However, all known algorithms for finding solutions take, for difficult examples, time that grows exponentially as the grid gets bigger.

So, Sudoku is in **NP** (quickly checkable) but does not seem to be in **P** (quickly solvable). Thousands of other problems seem similar, in that they are fast to check but slow to solve. Researchers have shown that many of the problems in **NP** have the extra property that a fast solution to any one of them could be used to build a quick solution to any other problem in **NP**, a property called **NP**-completeness. Decades of searching have not yielded a fast solution to any of these problems, so most scientists suspect that none of these problems can be solved quickly. This, however, has never been proven.