Yash NC

181210062

Java Assignment

P VS NP

**What is P vs NP Problem ?**

P vs. NP deals with the gap between computers being able to quickly solve problems vs. just being able to test proposed solutions for correctness. As such, the P vs. NP problem is the search for a way to solve problems that require the trying of millions, billions, or trillions of combinations without actually having to try each one. Solving this problem would have profound effects on computing, and therefore on our society.

This is why the answer to the P vs. NP problem is so interesting to people. If anyone were able to show that P is equal to NP, it would make difficult real-world problems trivial for computers.P problems are easily solved by computers, and NP problems are not easily solvable, but if you present a potential solution it’s easy to verify whether it’s correct or not.

**What does ‘P’ & ‘NP’ Signify ?**

**‘P’**

Problems that we know an efficient algorithm for that is capable of producing a solution in polynomial time are classified as P problems => P means polynomial time, in this instance. This was obviously the first subset of problems we were able to classify: of all these problems out there, at least we managed to solve these over here. Things like sorting lists, balancing trees, encrypting data are all problems that we have efficient algorithms for and so belong to the subset P.

**‘NP’**

After sometime We found another subset of problems that P itself was a subset of, NP problems. The NP stands for nondeterministic polynomial time, its part of the foundational, Turing-era computer science that underpins every single modern computer. What you do need to know is that NP problems do not have a known algorithm that can produce a result in polynomial time.

**What is the difference Between P Problems and NP problems ?**

|  |  |
| --- | --- |
| **P Problem** | **NP problem** |
| These problems can be solved in polynomial time by deterministic Algorithms | These problems can be solved in Non - Deterministic polynomial time. |
| These problems can be solved and verified in polynomial time | Solutions of these problems cannot be found in polynomial time but if solutions are given they can be verified in polynomial time. |
| P is the subset of NP problems | NP problem is the superset of P Problems |
| All Problems are deterministic in nature | All Problems are not deterministic in nature |

**Examples of :**

P Problems :

1)Recognizing palindromes

2)String matching

3)Recognizing relatively prime integers

4)Addition of two numbers

5)Multiplication of two numbers

NP Problems :

1)Knapsack problem.

2)Boolean satisfiability problem

3)Vertex cover

4)Independent set problem

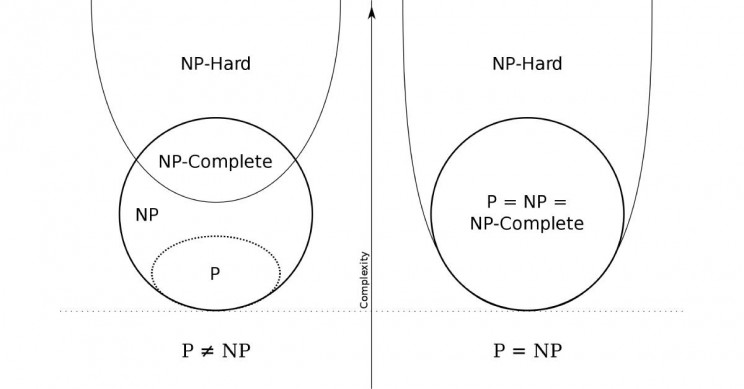
5)Graph coloring problem

**NP Hard & NP Completeness**

A NP-complete is a special category of NP problems that have time complexities greater than polynomial time, are verifiable in polynomial time, and belong to a set of problems known as NP-hard. NP-hard problems are essentially those that are at least as hard as the hardest NP problem, but don’t need to be verifiable in polynomial time.

Enumerating all the possible subsets of the set of every individual atom in the universe is an NP-hard problem. We cannot prove that such a problem is unsolvable in polynomial time, but there’s no reason to believe that we’ll ever find that algorithm

A problem is in the class NPC if it is in NP and is as hard as any problem in NP. A problem is NP-hard if all problems in NP are polynomial time reducible to it, even though it may not be in NP itself.



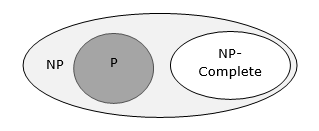
Another NP-hard problem is identifying a chess move in any given board state that is the absolute best move that you could make. In order determine this, you would have to know that every other move will lead to a worse outcome, and the only way that we know how to determine that is to follow every branching path of every move, countermove, and so on that is possible with the given board position. Once you arrive at the end result of each branch of this practically infinite decision tree, you would then take the best result and say that this was the very best move you could have made.

Definition of NP-Completeness

A language B is NP-complete if it satisfies two conditions :

1)B is in NP

2)Every A in NP is polynomial time reducible to B.

If a language satisfies the second property, but not necessarily the first one, the language B is known as NP-Hard. Informally, a search problem B is NP-Hard if there exists some NP-Complete problem A that Turing reduces to B. The problem in NP-Hard cannot be solved in polynomial time, until P = NP. If a problem is proved to be NPC, there is no need to waste time on trying to find an efficient algorithm for it

Examples of NP Hard and NP Complete Problems :

## NP-Complete Problems :

1)Determining whether a graph has a Hamiltonian cycle

2)Determining whether a Boolean formula is satisfiable.

## 

## NP-Hard Problems :

1)The circuit-satisfiability problem

2)Set Cover

3)Vertex Cover

4)Travelling Salesman Problem

**REAL LIFE EXAMPLES**

I) 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. 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 but does not seem to be in P.

II) Traveling salesman

A traveling salesman wants to visit 100 different cities by driving, starting and ending his trip at home. He has a limited supply of gasoline, so he can only drive a total of 10,000 kilometers. He wants to know if he can visit all of the cities without running out of gasoline.The traveling salesman problem consists of a salesman and a set of cities. The salesman has to visit each one of the cities starting from a certain one and returning to the same city. The challenge of the problem is that the traveling salesman wants to minimize the total length of the trip.This Problem is a NP complete problem

III) Farmer Problem

Farmer wants to take 100 watermelons of different masses to the market. She needs to pack the watermelons into boxes. Each box can only hold 20 kilograms without breaking. The farmer needs to know if 10 boxes will be enough for her to carry all 100 watermelons to market.