

NATIONAL INSTITUTE OF TECHNOLOGY

DELHI



Assignment -5

P v/s NP Problems

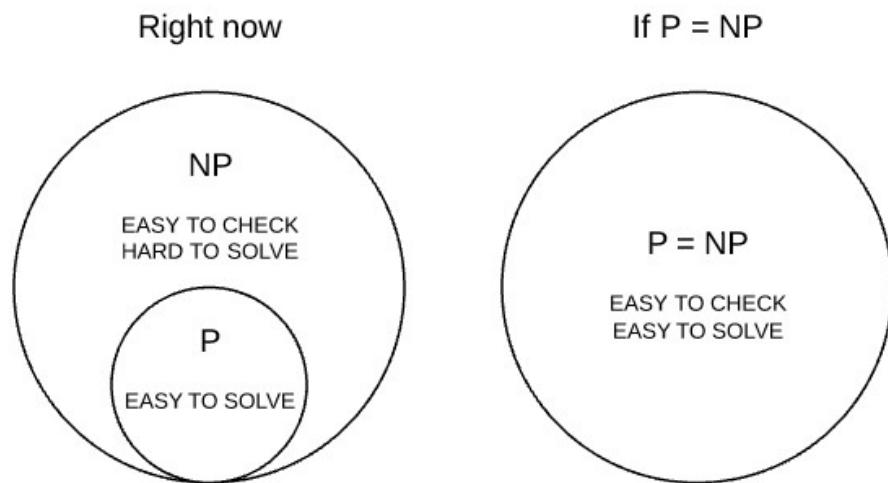
Vishal Nagargoje

181210059

CSE 2nd Year

Classifying Problems: P Vs NP

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, and encrypting data are all problems that we have efficient algorithms for and so belong to the subset P. Later, we found another subset of problems that P itself was a subset of, NP problems. The NP stands for nondeterministic polynomial time, but for our purposes, you don't need to know too much about what that means except that it's 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. However, if you are given a solution to an **NP problem**, verifying that it is correct is easy and can be done in **polynomial time** or less. We use this fact every time we unlock our iPhones or send messages over WhatsApp. As it turns out, **NP problems** are perfect for encryption; there is only one way to solve the problem that unlocks the encryption quickly, you need to have the answer ahead of time.



NP HARD & NP COMPLETE

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. If a polynomial time algorithm exists for any of these problems, all problems in NP would be polynomial time solvable. These problems are called NP-complete.

NP – Hard:

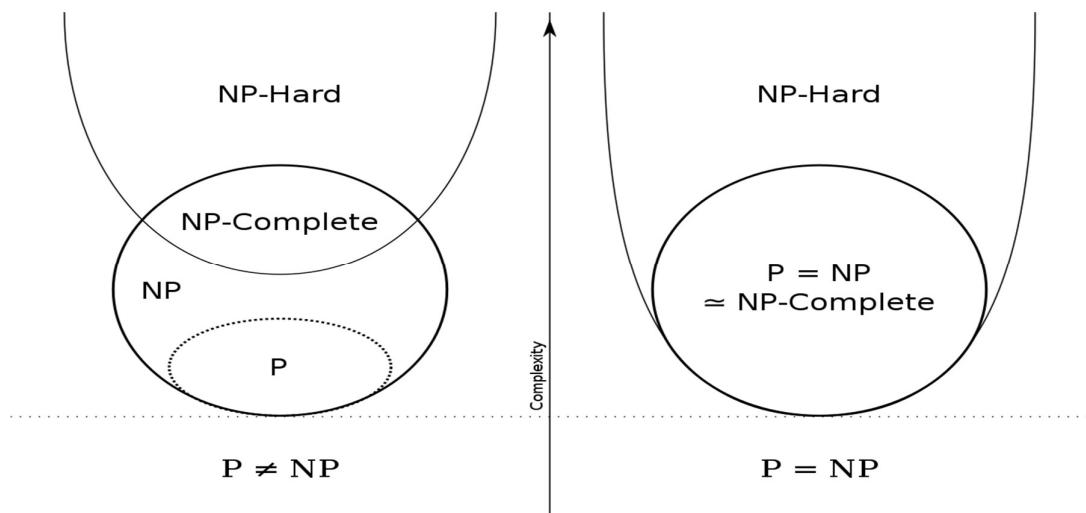
This is the class of problems which are at least as hard as the hardest problems in NP. Problems belonging to this class may or may not be part of NP, that is, the hardest problems of NP belong to the intersection of NP and NP-Hard. Problems in NP-Hard may not even be decision problems.

Example of a problem which is NP-Hard but not NP is the problem of identifying a chess move in any given board state that is the best possible move to make.

NP – Complete:

This is the class problems which contains the set of all the hardest problems in NP. Every problem in NP-Complete must belong to NP and NP-Hard, which is not true for NP-Hard. NP-Complete is the intersection of NP and NP-Hard.

Example of a problem which is NP-Complete is the clique graph problem, where, in an undirected graph, the largest complete sub-graph is to be found.



EXAMPLE:

