Ans. To. The Q. No. 08

NP-hard:-

An NP-hard problem is at least as hard as a NP-complete problem. However, when a computer scientist says that a problem is NP-hard, it usually imply that it is strictly harder than NP-complete. Typically, the problem cannot be solved by a Known deterministic polynomial algorithm, AND when a solution is given, one can't determine whether the solution is correct in polynomial time.

Example

The halting problem is an NP-hard problem. This is the problem that given a program P and input I, will it halt? This is a dicision problem it is not in NP. It is clear that any NP-complete problem can be reduced to this one. As another example, any NP-complete problems is NP-hard.

NP-Complete:-

An NP-Complete problem is a problem that can't be solve by a Known deterministic polynomial algorithm, but can be solve by a non-deterministic polynomial algorithm. Purely non-deterministic system do not actually exist, but that means that if a solution is given, we can easily decide wheather the solution is correct or not.

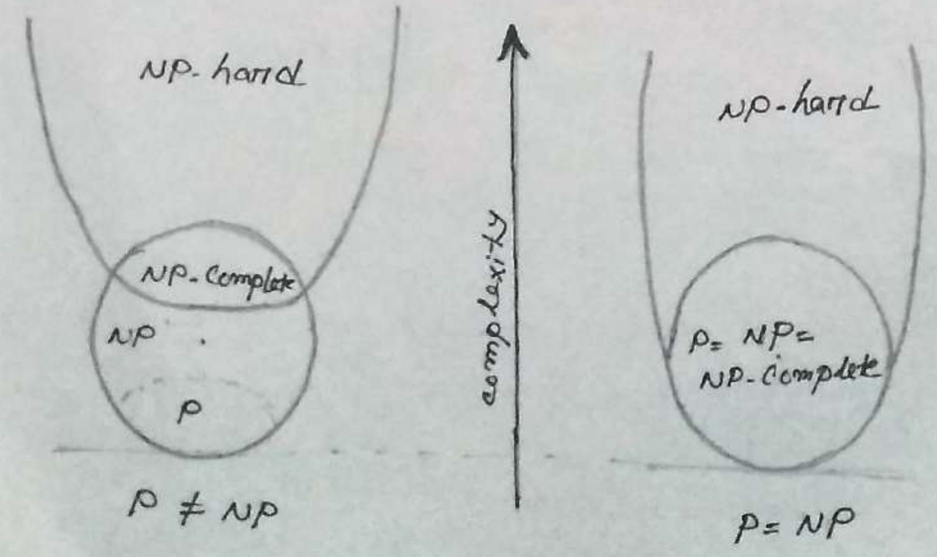
Example

3-SAT. This is the problem wherein we are given a conjunction (ANDs) of 3-clause disjunctions (ORs), statements of the form.

```
(x-v11 OR x-v21 OR x-v31) AND
(x-v12 OR x-v22 OR x-v32) AND
......                     AND
(x-v1n OR x-v2n OR x-v3n)
```

where each x-vij is a boolean variable or the negation of a variable from a finite predefined list (x-1, x-2, ... x-n).

* here is the typical schema, to show the difference between NP-hard and NP-complete :-



* Difference between NP-hard and NP-complete:-

NP-hard	NP-complete
1) NP-hard problem (say X) can be solved if and only if there is a NP-complete problem (say Y) can be reducible into X in polynomial time.	1) NP-complete problems can be solved by deterministic algorithm in polynomial time.
2) To solve this problem, it must be a NP-problem.	2) To solve this problem, it must be both NP and NP-hard problem.
3) It is not a decision problem.	3) It is exclusively decision problem.
4) Example: Halting problem, vertex cover problem, circuit satisfiability problem etc.	4) Example: Determine whether a graph has a Hamiltonian cycle, Determine whether a Boolean formula is satisfiable or not, etc.