know the consequences of being in NP and NP-complete

NP = non-deterministic polynomial

- it allows a choice of next-step
- uses yes/no PROBLEMS

↓

accepts

Time complexity = polynomial

$O(n^2)$ / $O(n^k)$

n = input size
k = constant (like 2)

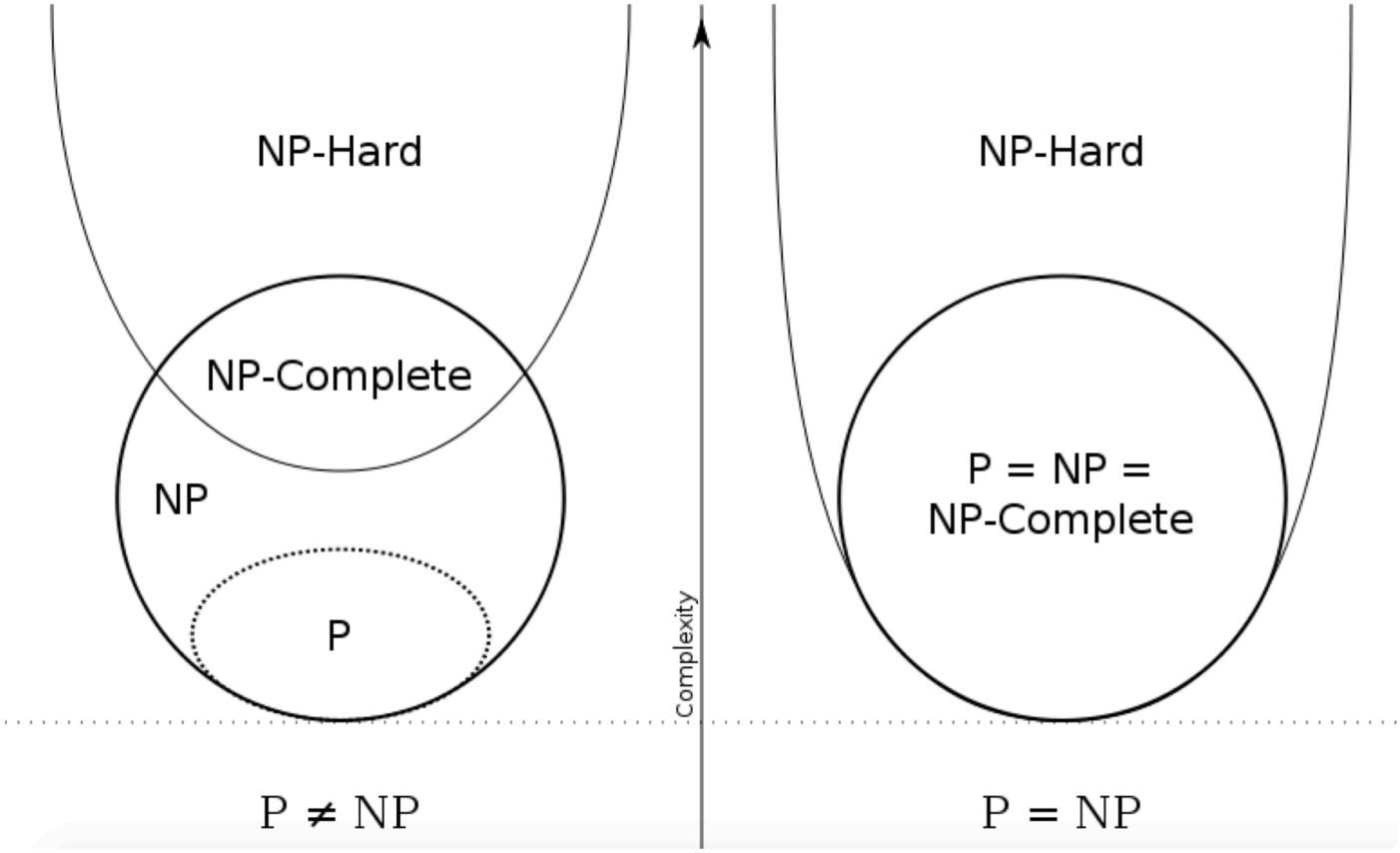
if F is satisfiable, there exists a way for this algorithm to reach an accepting state

```
nondeterministic algorithm
 for i = 1 to n
    set x[i] = 0 or 1 (nondet step)

if F[x[1],…,x[n] is true
   then ACCEPT
   else REJECT
```

NP-Complete: If one NP-complete problem can be solved in poly-time, then all of NP can be solved in poly-time

↑ no one knows if this is possible but if you prove one, you prove it all.



QUESTIONS:

1) If a single NP-complete problem is shown to have a polynomial time (deterministic) algo, then it can be concluded that P=NP

TRUE. If you prove one, you prove all.

2) Whether any NP-complete problem is P OPEN.

3) If a single NP-complete problem is shown to require exponential time, then it can be concluded that P=NP.

FALSE

4) Every problem that can be solved in polynomial time non-deterministically can be solved by a polynomial time deterministic algo. OPEN

5) There is an NP-complete problem with a poly-time algo. OPEN

6) If one NP-complete problem is proved to require exponential time, then P=NP. FALSE

7) If one NP-complete problem is proved to require exponential time, then P!=NP. TRUE