CS 218 Design and Analysis of Algorithms

Nutan Limaye

Indian Institute of Technology, Bombay
nutan@cse.iitb.ac.in

Module 3: NP hardness and reductions

Are all problems efficiently solvable?

In Module 1 and 2 we saw many problems that are efficiently solvable.

One might feel that indeed all problems ARE efficiently solvable.

Is this true?

The answer is we do not know.

Are all problems efficiently solvable?

There are problems believed to be impossible to solve efficiently.

Unfortunately, many of them are extremely important problems.

Which we would like to solve efficiently.

In this module we will see some examples of such problems.

We will also build a theory around such problems.

P and NP

Definition

A problem Π is said to be in the class P if there exists an algorithm $\mathcal{A}$ such that for any input x, $\mathcal{A}$ finds the correct solution for x in time poly($|x|$) time.

Examples

- Searching, sorting.
- Interval scheduling.
- Integer multiplication, GCD computation.
- Several string related optimization problems.
- Max-flow, min-cut, bipartite perfect matching.
- Primality testing, perfect matching in general graphs, ...

P and NP

For problems in NP, let us start with some examples.

3-colorability problem

Given: $G = (V, E)$

Check: Can V be colored with 3 colors such that for all $e = (u, v) \in E$, color of u is not equal to color of v.

Brute-force algorithm will need $O(3^n)$ time.

If $n = 3000$ then will take too long to solve.

P and NP

For problems in NP, let us start with some examples.

3-colorability problem

Given: $G = (V, E)$
Check: Can V be colored with 3 colors such that for all $e = (u, v) \in E$, color of u is not equal to color of v.

Suppose, someone magically provides the coloring $c : V \to \{\text{R,G,B}\}$

In polynomial time can check whether it is or it is not.

```
Test-Coloring(G, c)
    For each e = (u, v), check if c(u) ≠ c(v).
    If all checks succeed then output 'Yes'.
    Else output 'No'.
```

If the graph is 3-colorable, then there exists a coloring c that makes `Test-Coloring` output `Yes`.

If the graph is not 3-colorable, then for all colorings, `Test-Coloring` outputs `No`.

P and NP

k-clique problem

Given: $G = (V, E)$

Check: Does there exists a subset $S \subseteq V$ such that $|S| = k$ and $\forall u, v \in S$, $(u, v) \in E$.

Brute-force algorithm will need $O(n^k)$ time.

Will take too long if k is growing, say e.g. $k = \sqrt{n}$.

P and NP

k-clique problem

Given: $G = (V, E)$
Check: Does there exists a subset $S \subseteq V$ such that $|S| = k$ and $\forall u, v \in S$, $(u, v) \in E$.

Suppose, someone magically provides a subset $S \subseteq V$

In polynomial time can check whether it is a clique or not.

```
Test-Clique(G, S of size k)
    For each u,v ∈ S, check if (u,v) ∈ E.
    If all checks succeed then output ‘Yes’.
    Else output ‘No’.
```

If the graph has a k-clique, then there exists a subset S that makes `Test-Clique` output `Yes`.

If the graph has no k-clique, then for all subsets of size k, `Test-Clique` outputs `No`.

P and NP

Satisfiability problem

Given: a CNF formula ϕ over variables $x_1, \ldots, x_n$
Check: does there exists an assignment $\tilde{a} \in \{0,1\}^n$ to $x_1, \ldots, x_n$ such that $\tilde{a}$ satisfies ϕ.

The brute-force algorithm will take time $O(2^n)$.

It will take too long if n is 2000 or so.

P and NP

Satisfiability problem

Given: a CNF formula ϕ over variables $x_1, \ldots, x_n$
Check: does there exists an assignment $\tilde{a} \in \{0,1\}^n$ to $x_1, \ldots, x_n$ such that $\tilde{a}$ satisfies ϕ.

If ϕ is satisfiable, then there exists an assignment $\tilde{a}$ that witnesses this.

If ϕ is not satisfiable, then for all assignments of $x_1, \ldots, x_n$, ϕ will not evaluate to 1.

P and NP

We now define NP.

Definition

A problem Π is said to be in NP if there is a polynomial time algorithm $\mathcal{T}$ and the following conditions hold.

- If input x is a positive instance of Π then there is a polynomial length proof y such that $\mathcal{T}$ on inputs x, y outputs 'Yes'.
- If input x is a negative instance of Π then for any polynomial length proof y, $\mathcal{T}(x, y)$ outputs 'No'.

Is every problem in P also in NP?

Does not need a proof.