15750  Lecture 20  (NP Completeness and Hardness of Problems)

\[ \text{max } c^T x \]
\[ \text{subject to } Ax \leq b \]
\[ x \geq 0 \]
\[ x \in \text{integer} \quad \forall i \]
\[ x \in \{0,1\} \quad \forall i \]

\[ \text{ILP} \]

\[ \text{Inty ILP} \]

\[ \text{Thm} 22: \text{Algorithm for Integer Linear Programming that runs efficiently (unless P = NP)} \]

\[ \text{Sad news: possible (not probable) that every algorithm for ILP runs in time } O(n) \]
1. Polynomial Time < Efficiency
2. Decision Problems (vs. Search Problems)
3. Witnesses and NP
4. Reductions
5. NP Completeness
II. Decision Problems

Problem: input
only allowed answers/outputs are YES/NO.

GRAPH CONNECTIVITY: Input: Graph G
Output: YES G is connected
NO otherwise

BIPARTIENESS: Input: G
Output: is G bipartite?

VS. SEARCH

PRIMALITY: Input: n
Output: is n prime?

SHORTEST PATH: Input: G, edge lengths, s, t, K
Output: length of shortest s-t path
is length of shortest s-t path ≤ K?
(3) **Polynomial Time**: Given any problem which takes input $I$:

- A runs in poly time if it constant $c > 0$ such that for an input $I$, $A(I)$ runs for time $\leq |I|^c$ and outputs correct answer.

**Shortest path on Graph**

- $I = (n, m)$
- $(u_1, v_1, e_1), \ldots, (u_m, v_m, e_m)$
  - $n$: number of vertices
  - $m$: number of edges
  - $e_i$: edge $(u_i, v_i)$

**Length of path**

- $2\log n + O(\log L)$ bits + $O(\log L)$ bits $\times m$

**Dijkstra's Algorithm**

- Takes $O(m \log n)$ operations on $\#s \leq nL$
- Each arithmetic op takes $O(\log (nL))$ time

**Summary**

- Total time complexity:
  - $O(m \log n + m \log L)$
Primality \((n):\)

\[
\text{for } i = 1 \ldots \sqrt{n} \text{ do }
\begin{cases}
\text{check if } i \text{ divides } n & \leftarrow \text{say takes } O(1) \text{ time} \\
\text{if so, say } n \text{ not prime, stop}
\end{cases}
\]

say "\(n\) prime"

\text{runtime } = O(\sqrt{n})

Input: keep \(I\) = \log n

\text{runtime} = \sqrt{n} = 2^{\frac{1}{2} \log n} = 2^{\frac{1}{2} |I|} \leftarrow \text{exponential}

\text{Best runtime for factoring } = 2^{\left(\log n\right)^{\frac{1}{3}}}

$P = \exists$ prob decision problems that have polytime algos

$NP \neq$ poly time

problems whose "solutions can be easily verified"

Decision Problem $Q$ is in $NP$ if \exists an algo $V$ called verifier such that:

- $V(I)$ instance

  $\exists$ a possibly nonunique witness $W$ of poly (|I|) size

  $\begin{align*}
  \text{if } \text{answer}(I) = \text{YES} \text{ then } \exists \text{ witness } W \text{ of poly (|I|) size} \\
  \text{for which } V(I, W) \text{ says } \text{OK}
  \end{align*}$

- $\text{if } \text{answer}(I) = \text{NO} \text{ then } \forall$ possible strings $W$ of length poly (|I|)

  $V(I, W) \text{ says } \text{NOpe!}$
Example:

3COLORABLE: Input Graph G
Output: YES if there is a way to color the vertices of G
such that no adjacent vertices get the same color.
**COMPOSITENESS**

Input number \( n \)

Output is **YES** if \( n \) is not prime

**NO** if \( n \) is prime

**Witness** = \( a, b \), \( a, b \neq 1 \), and \( a \cdot b = n \)

**Verify** = check if

**PRIMES**

Input \( n \)

Output:

**YES** if \( n \) is a prime

**NO** if \( n \) is composite

\[ \Rightarrow \text{COMPOSITENESS} \in \text{NP} \]

\[ \Rightarrow \text{PRIMES} \in \text{NP} \]

\[ \Rightarrow \text{PRIMES} \in \text{P} \Rightarrow \text{PRIMES} \in \text{NP} \]
**NP Complete Problems:** The "hardest" Problems in NP

Q is NP-complete if

(a) Q ∈ NP
(b) if Q has a poly-time algo
   \[ \Rightarrow \exists Q' \in NP, \ Q' \text{ has a poly-time algo.} \]

Q is NP-complete if

(a) Q ∈ NP
(b) \[ \exists \text{ another NP complete problem } Q', \ st. \ Q \text{ has a poly-time algo.} \]
\[ \Rightarrow Q' \text{ has a poly-time algo.} \]

(c) \( Q = \text{3COLORING} \)
Assert 3 COLORING is a hardest problem in NP

3COLORING in NP complete

Steps: Want to show 4 COLORING is NP complete

1. 4 COLORING ∈ NP ← Linear time verification
2. 4 COLORING ∈ P → 3COLORING ∈ P

⇒ ∃ algo A that given a graph G, color using 4 colors in time \( n^c \), \( n = |V| \)

⇒ 3COLORING is no harder than 4 coloring
Coloring with 1V1 colors is easy.