

CSE 105—Theory of Computability

Lecture 19—June 6, 2006
Time Complexity

Instructor: Neil Rhodes

Review of Big-Oh notation

Let f and g be functions $f, g: \mathbb{N} \rightarrow \mathbb{R}^+$. $f(n) = O(g(n))$ if

- There exist positive integers c, n_0 such that,
- for all $n \geq n_0$:
 - $f(n) \leq c \cdot g(n)$

$g(n)$ is an *asymptotic upper bound* for $f(n)$

Time and Space complexity

$\text{TIME}(f(n)) = \{L \mid L \text{ is decidable by a deterministic Turing machine in } O(f(n)) \text{ steps}\}$

$\text{SPACE}(f(n)) = \{L \mid L \text{ is decidable by a deterministic Turing machine that reads } O(f(n)) \text{ tape cells}\}$

$\text{NTIME}(f(n)) = \{L \mid L \text{ is decidable by a non-deterministic Turing machine in } O(f(n)) \text{ steps}\}$

- How many steps does a non-deterministic TM take on a particular input?

$\text{NSPACE}(f(n)) = \{L \mid L \text{ is decidable by a non-deterministic Turing machine that reads } O(f(n)) \text{ tape cells}\}$

The class P

The set of languages which are recognizable by a **deterministic** Turing machine in polynomial time

- Union of $\text{TIME}(n^k)$ for all k

The class NP

The set of languages which are recognizable by a **deterministic** Turing machine in polynomial time

- Union of $\text{NTIME}(n^k)$ for all k
- Alternate definition: NP is the set of languages which have a polynomial-time *verifier*

Open Question

Is $P = NP$?

- Clearly, $P \subseteq NP$

NP-Complete

There exist problems which are as hard as any in NP (**NP-hard**).

- There is a polynomial time reduction from any problem in NP to an NP-Hard problem

If such a problem is also in NP, then it is **NP-complete**

- If you find a polynomial time algorithm for *any* NP-complete problem, all NP-complete problems have a polynomial-time algorithm

The first NP-complete problem

- 3-SAT: Given a boolean formula in CNF:
 - $(x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \wedge x_4 \wedge \neg x_3) \wedge \dots (x_3 \vee x_2 \vee x_1)$
- Is there an assignment of truth values that satisfies the formula

Example NP-Complete Problem

Vertex-cover is NP-complete

- Vertex-cover = $\{ \langle G, k \rangle \mid G \text{ is an undirected graph that has a } k\text{-node vertex cover} \}$

To show:

- Vertex-cover $\in$ NP
- 3-SAT $\leq$ Vertex-cover
- Example 3-SAT formula:
 - $(x_1 \vee x_2 \vee x_3) \wedge (\sim x_1 \wedge x_3 \wedge \sim x_2) \wedge (x_1 \vee \sim x_2 \vee \sim x_3)$

EXPTIME

Exponential time:

- $\text{EXPTIME} = \{L \mid L \text{ is decidable by a deterministic TM in } \text{TIME}(2^{n^k}) \text{ for some } k\}$
- Relationship between:
 - P
 - NP
 - EXPTIME