# COMPSCI 130B Discussion

01/07/2021

Kha-Dinh (Jacob) Luong

Office hours: F 9-11 AM

# Objectives

- Time complexity analysis.
- Proof by induction.

# Time Complexity Analysis

- Evaluate the running time of a procedure/algorithm.
- Expressed in term of a function of the input size.
- Asymptotic notations Big O, Big $\Omega$, Big $\Theta$.

# Time Complexity Analysis – Big O

- Read: Asymptotic Upper Bound.
- $f(n) = O(g(n)) \Leftrightarrow \forall n > n_0, \exists c > 0, f(n) \leq c * g(n)$.
- f(n) grows no faster than g(n).

# Time Complexity Analysis – Big Ω

- Read: Asymptotic Lower Bound.
- $\mathrm{f}(n) = \Omega(\mathrm{g(n)}) \Leftrightarrow \forall \mathrm{n} > \mathrm{n}_0, \exists \mathrm{c} > 0, \mathrm{c} * \mathrm{g}(n) \leq \mathrm{f}(n).$
- f(n) grows no slower than g(n)

# Time Complexity Analysis – Big Θ

- Read: Composite bound.
- $f(n) = \Theta(g(n)) \Leftrightarrow f(n) = O(g(n))$ and $f(n) = \Omega(g(n))$.
- f(n) grows at the same rate as g(n).

# Time Complexity Analysis

- Small *o* and small $\omega$: very similar to Big O and Big $\Omega$.
- $f(n) = o(g(n)) \Leftrightarrow \forall n > n_0, \exists c > 0, f(n) < c * g(n)$.
- $f(n) = \omega(g(n)) \Leftrightarrow \forall \mathrm{n} > \mathrm{n}_0, \exists \mathrm{c} > 0, \mathrm{c} * \mathrm{g}(n) < \mathrm{f}(n)$.

# Time Complexity Analysis

$$\begin{aligned} f = O(g) \text{ and } f = \Omega(g) &\Leftrightarrow f = \Theta(g) \\ f = O(g) &\Leftrightarrow g = \Omega(f) \\ f = o(g) &\Leftrightarrow g = \omega(f) \end{aligned}$$

$$\begin{aligned} f = o(g) &\Rightarrow f = O(g) \\ f = \omega(g) &\Rightarrow f = \Omega(g) \\ f \sim g &\Rightarrow f = \Theta(g) \end{aligned}$$

$$\begin{aligned} \lim_{n\to\infty} f(n)/g(n) \neq 0, \infty &\Rightarrow f = \Theta(g) \\ \lim_{n\to\infty} f(n)/g(n) \neq \infty &\Rightarrow f = O(g) \\ \lim_{n\to\infty} f(n)/g(n) \neq 0 &\Rightarrow f = \Omega(g) \end{aligned}$$

$$\begin{aligned} \lim_{n\to\infty} f(n)/g(n) = 1 &\Rightarrow f \sim g \\ \lim_{n\to\infty} f(n)/g(n) = 0 &\Rightarrow f = o(g) \\ \lim_{n\to\infty} f(n)/g(n) = \infty &\Rightarrow f = \omega(g) \end{aligned}$$

http://web.mit.edu/broder/Public/asymptotics-cheatsheet.pdf

# Time Complexity Analysis

- Generally, time complexity is determined by the most dominant term in the function.
  - $O(5) = O(1)$
  - $O(2n^2) = O(n^2)$
  - $O(4n^3 + 2n^2 + 1) = O(n^3)$
  - $O(2n + 5\log(n)) = O(n)$
  - $O(nlog(n) + n) = O(nlog(n))$
- In terms of dominance:

$$a^n > n^a > log_a n > a$$

# Time Complexity Analysis - Example

- Prove that $n^2 + 17n + 2$ is $O(n^2)$.

$$\begin{aligned} n^2 + 17n + 2 &\le n^2 + 17n^2 + 2 \text{ (when n} \ge 2) \\ &\le n^2 + 17n^2 + n^2 \text{ (when n} \ge 2) \\ &= 19n^2 \end{aligned}$$

So when $n \ge 2$, with $c = 19$, $n^2 + 17n + 2 \le cn^2$. Therefore, $n^2 + 17n + 2 = O(n^2)$.

- Is $(n^2+17n + 2)$ $O(n^3)$? $O(n)$?

# Time Complexity Analysis - Example

- Prove that $log_{10}(x)$ is $O(log_2(x))$.

$$\lim_{n\to\inf} \frac{\log_{10}(x)}{\log_2(x)} = \lim_{n\to\inf} \frac{\log_2(x)}{\log_2(10)\log_2(x)} = \lim_{n\to\inf} \frac{1}{\log_2(10)} = \frac{1}{\log_2(10)}$$

Therefore, $\log_{10}(x) = \Theta(log_2(x))$, which implies $\log_{10}(x) = O(log_2(x))$.

# Proof by Induction

- Prove an assertion $P(n)$ is true for all $n \geq 0$.

- Base cases:
  - Check that $P(n)$ holds for $0 \leq n \leq k$.

- Induction step:
  - Prove that $P(n) \Rightarrow P(n+1)$ for $n > k$.

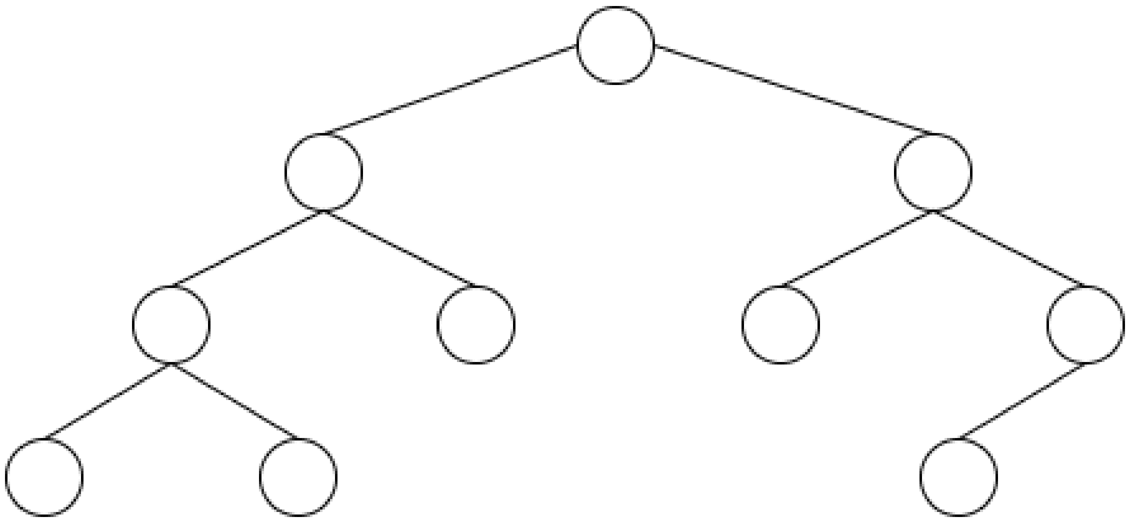
- Conclusion: $P(n)$ is true for all $n \geq 0$.

# Proof by Induction - Example

- $P(n)$: $7^n - 1 \vdots 6$
- Base case: $n = 1, 7^1 - 1 = 6 \vdots 6$
- Induction step:
  - Assume $P(k)$ holds.
  - $7^{k+1} - 1 = 7\left(7^k - 1\right) + 6 \vdots 6.$
  - Therefore, $P(k + 1)$ also holds.
- Conclusion: $P(n)$ holds for all $n \geq 1$.

# Proof by Induction - Example

- $P(n)$: A binary tree with $n$ nodes has $n-1$ edges.



# Proof by Induction - Example

- Base case: Binary tree with 1 node has 0 edge.
- Induction step:
  - Assume $P(k)$ is true.
  - Consider a tree with $k + 1$ nodes.
  - Hint: Removing a random leaf node.
  - ...
- Conclusion.