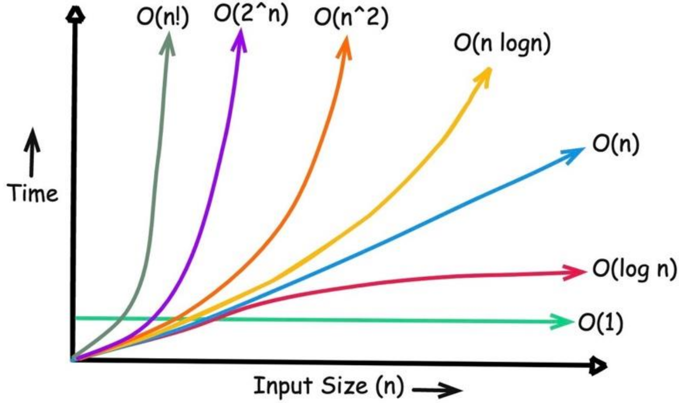
Lecture 4

Algorithm Analysis III



Announcements

- Homework 1 is graded
- Due Sunday night @ 11:59pm:
  - Reflections on Homework 1
  - Homework 2
- Group Meetings start next week
  - Groups will be posted over the weekend
  - Self-scheduled meeting for an hour studying, working on HW, completing practice exercises
- Start thinking about Individual Project 1
  - Due 2/27

Algorithm Design

1. Formulate the problem precisely
2. Design an algorithm
3. Prove the algorithm is correct
4. Analyze its running time

Revisiting Big-O

Definition: The function $T(n)$ is $O\big(f(n)\big)$ if there exists constants $c > 0$ and $n_0 \geq 0$ such that

$$T(n) \leq cf(n) \text{ for all } n \geq n_0$$

We say that f is an **asymptotic upper bound** for T

Revisiting Big-O

Definition: The function $T(n)$ is $O\big(f(n)\big)$ if there exists constants $c > 0$ and $n_0 \geq 0$ such that

$$T(n) \leq cf(n) \text{ for all } n \geq n_0$$

We say that f is an **asymptotic upper bound** for T

One limitation is that an asymptotic upper bound may not be **tight**

- $T(n) = 2n^2 + n \log n$ is $O(n^2), O(n^3), \dots, O\big(n^k\big), \dots$

Revisiting Big-O

Definition: The function $T(n)$ is $O\big(f(n)\big)$ if there exists constants $c > 0$ and $n_0 \geq 0$ such that

$$T(n) \leq cf(n) \text{ for all } n \geq n_0$$

We say that f is an **asymptotic upper bound** for T

One limitation is that an asymptotic upper bound may not be **tight**

- $T(n) = 2n^2 + n\log n$ is $O(n^2), O(n^3), \dots, O\big(n^k\big), \dots$
- What would it mean for a bound to be tight?

Exercise I

Suppose f is $O(g)$. Which of the following is true?

i. g is $O(f)$
ii. g is not $O(f)$
iii. g may be $O(f)$, depending on f and g

Exercise I

Suppose f is $O(g)$. Which of the following is true?

i. g is $O(f)$

ii. g is not $O(f)$

iii. g may be $O(f)$, depending on f and g

Big-Ω Definition

Informally: T grows at least as fast as f

Definition: The function $T(n)$ is $\Omega\big(f(n)\big)$ if there exists constants $c > 0$ and $n_0 \geq 0$ such that

$$T(n) \geq cf(n) \text{ for all } n \geq n_0$$

We say that f is an **asymptotic lower bound** for T

Big-Ω Definition

Informally: T grows at least as fast as f

Definition: The function $T(n)$ is $\Omega\big(f(n)\big)$ if there exists constants $c > 0$ and $n_0 \geq 0$ such that

$$T(n) \geq cf(n) \text{ for all } n \geq n_0$$

We say that f is an **asymptotic lower bound** for T

What's the difference between Big-O and Big-Ω?

Big-Ω Examples

$T(n) = 4n + 10$ is $\Omega(n)$

$T(n) = \frac{1}{2}\, n^2$ is $\Omega(n^2)$

Exercise II

Claim: $n - 10$ is $\Omega(n)$

To prove this, we need to show that $n - 10 \geq cn$ for all $n \geq n_0$.

Q: What is the largest value of c below for which we can find some n_0 to make this statement true?

i. $c = 0.5$

ii. $c = 0.99$

iii. $c = 2$

iv. $c = 20$

Exercise II

Claim: $n - 10$ is $\Omega(n)$

To prove this, we need to show that $n - 10 \geq cn$ for all $n \geq n_0$.

Q: What is the largest value of c below for which we can find some n_0 to make this statement true?

i. $c = 0.5$

ii. $c = 0.99$

iii. $c = 2$

iv. $c = 20$

Big-Θ Definition

Informally: T grows just as fast as f

Definition: The function $T(n)$ is $\Theta\big(f(n)\big)$ if $T(n)$ is both $O\big(f(n)\big)$ and $\Omega\big(f(n)\big)$

We say that f is an **asymptotically tight bound** for T

Big-Θ Definition

Informally: T grows just as fast as f

Definition: The function $T(n)$ is $\Theta(f(n))$ if $T(n)$ is both $O(f(n))$ and $\Omega(f(n))$

We say that f is an **asymptotically tight bound** for T

Exercise III

Suppose $T(n) = 32n^2 + 17n + 2$. Which of the following statements are true?

i. $T(n)$ is $\Theta(n)$

ii. $T(n)$ is $\Theta(n^2)$

iii. $T(n)$ is $\Theta(n^3)$

Exercise III

Suppose $T(n) = 32n^2 + 17n + 2$. Which of the following statements are true?

i. $T(n)$ is $\Theta(n)$

ii. $T(n)$ is $\Theta(n^2)$

iii. $T(n)$ is $\Theta(n^3)$

Exercise III

Suppose $T(n) = 32n^2 + 17n + 2$. Which of the following statements are true?

i.	$T(n)$ is $\Theta(n)$	$T(n)$ is $O(n)$	$T(n)$ is $\Omega(n)$
ii.	$T(n)$ is $\Theta(n^2)$	$T(n)$ is $O(n^2)$	$T(n)$ is $\Omega(n^2)$
iii.	$T(n)$ is $\Theta(n^3)$	$T(n)$ is $O(n^3)$	$T(n)$ is $\Omega(n^3)$

Exercise III

Suppose $T(n) = 32n^2 + 17n + 2$. Which of the following statements are true?

i.	$T(n)$ is $\Theta(n)$	~~$T(n)$ is $O(n)$~~	$T(n)$ is $\Omega(n)$
ii.	$T(n)$ is $\Theta(n^2)$	$T(n)$ is $O(n^2)$	$T(n)$ is $\Omega(n^2)$
iii.	$T(n)$ is $\Theta(n^3)$	$T(n)$ is $O(n^3)$	~~$T(n)$ is $\Omega(n^3)$~~

"Good" Running Time

Inefficiency

- ❖ We said that 2^n steps or worse is unacceptable in practice
- ❖ i.e. $O(2^n)$ or exponential running time is inefficient

"Good" Running Time

Inefficiency

- ❖ We said that 2^n steps or worse is unacceptable in practice
- ❖ i.e. $O(2^n)$ or exponential running time is inefficient

Efficiency

- ❖ An algorithm is *efficient* if it has a polynomial running time
- ❖ i.e. $O\left(n^k\right), k \geq 0$

"Good" Running Time

Inefficiency

- ❖ We said that 2^n steps or worse is unacceptable in practice
- ❖ i.e. $O(2^n)$ or exponential running time is inefficient

Efficiency

- ❖ An algorithm is *efficient* if it has a polynomial running time
- ❖ i.e. $O\left(n^k\right), k \geq 0$

Exceptions

- ❖ Some poly-time algorithms have large constants and exponents
- ❖ We sometimes use exponential-time algorithms when their worst case does not arise in practice

Next Time

- Common running times
- Examples of problems for each class