Week 1 - Friday

COMP 2230

Last time

- Policies and course overview
- Graphing functions
- Big-O, Big-Omega, and Big-Theta notation

Questions?

Assignment 1

Logical warmup

- You have three pitchers whose capacities are three, four, and seven quarts
- Only the seven-quart pitcher is full
- Pour the fewest times to separate the water into two-, two-, and three-quart quantities

Asymptotic Notations Continued

Definitions

- Let f and g be real-valued functions defined on the same set of nonnegative real numbers
- f **is of order at least** g, written $f(x)$ is $\Omega(g(x))$, iff there is a positive $A \in \mathbb{R}$ and a nonnegative $a \in \mathbb{R}$ such that
  - $A|g(x)| \leq |f(x)|$ for all $x > a$
- f **is of order at most** g, written $f(x)$ is $\mathrm{O}(g(x))$, iff there is a positive $B \in \mathbb{R}$ and a nonnegative $b \in \mathbb{R}$ such that
  - $|f(x)| \leq B|g(x)|$ for all $x > b$
- f **is of order** g, written $f(x)$ is $\Theta(g(x))$, iff there are positive $A, B \in \mathbb{R}$ and a nonnegative $k \in \mathbb{R}$ such that
  - $A|g(x)| \leq |f(x)| \leq B|g(x)|$ for all $x > k$

Using the notation

- Express the following statements using appropriate notation:
  - $10|x^6| \leq |17x^6 - 45x^3 + 2x + 8| \leq 30|x^6|$, for $x > 2$
  - $15\left|\sqrt{x}\right| \leq \left|\frac{15\sqrt{x}(2x+9)}{x+1}\right|, x > 0$
  - $\left|\frac{15\sqrt{x}(2x+9)}{x+1}\right| \leq 45\left|\sqrt{x}\right|, x > 6$
- Justify the following:
  - $\left|\frac{15\sqrt{x}(2x+9)}{x+1}\right|$ is $\Theta(\sqrt{x})$

Properties of Ω-, O-, and Θ-notation

- Let f, g, h, and k be real-valued functions defined on the same set of nonnegative real numbers

1. $f(x)$ is $\Omega(g(x))$ and $f(x)$ is $O(g(x))$ iff $f(x)$ is $\Theta(g(x))$
2. $f(x)$ is $\Omega(g(x))$ iff $g(x)$ is $O(f(x))$
3. $f(x)$ is $\Omega(f(x))$, $f(x)$ is $O(f(x))$, and $f(x)$ is $\Theta(f(x))$
4. If $f(x)$ is $O(g(x))$ and $g(x)$ is $O(h(x))$ then $f(x)$ is $O(h(x))$
5. If $f(x)$ is $O(g(x))$ and c is a positive real, then $c \cdot f(x)$ is $O(g(x))$
6. If $f(x)$ is $O(h(x))$ and $g(x)$ is $O(k(x))$ then $f(x) + g(x)$ is $O(G(x))$ where $G(x) = \max(|h(x)|, |k(x)|)$ for all x
7. If $f(x)$ is $O(h(x))$ and $g(x)$ is $O(k(x))$ then $f(x) \cdot g(x)$ is $O(h(x) \cdot k(x))$

Orders of functions

- If $1 < x$, then
  - $x < x^2$
  - $x^2 < x^3$
  - …
- So, for $r, s \in \mathbb{R}$, where $r < s$ and $x > 1$,
  - $x^r < x^s$
  - By extension, $x^r \; is \; O(x^s)$

Proving bounds

- Prove a Θ bound for $g(x) = \frac{1}{4}(x - 1)(x + 1)$ for $x \in \mathbb{R}$
- Prove that x^2 is not $O(x)$
  - Hint: Proof by contradiction

Polynomials

- Let $f(x)$ be a polynomial with degree n
  - $f(x) = a_n x^n + a_{n-1} x^{n-1} + a_{n-2} x^{n-2} \ldots + a_1 x + a_0$
- By extension from the previous results, if a_n is a positive real, then
  - $f(x)$ is $O(x^s)$ for all integers $s \geq n$
  - $f(x)$ is $\Omega(x^r)$ for all integers $r \leq n$
  - $f(x)$ *is* $\Theta(x^n)$
- Furthermore, let $g(x)$ be a polynomial with degree m
  - $g(x) = b_m x^m + b_{m-1} x^{m-1} + b_{m-2} x^{m-2} \ldots + b_1 x + b_0$
- If a_n and b_m are positive reals, then
  - $f(x)/g(x)$ is $O(x^c)$ for real numbers $c > n - m$
  - $f(x)/g(x)$ is **not** $O(x^c)$ for real numbers $c < n - m$
  - $f(x)/g(x)$ is $\Theta(x^{n-m})$

Algorithm Efficiency

Three-Sentence Summary

Algorithm Efficiency

Extending notation to algorithms

- We can easily extend our Ω-, O-, and Θ- notations to analyzing the running time of algorithms
- Imagine that an algorithm A is composed of some number of elementary operations (usually arithmetic, storing variables, etc.)
- We can imagine that the running time is tied purely to the number of operations
- This is, of course, a lie
  - Not all operations take the same amount of time
  - Even the **same** operation takes different amounts of time depending on caching, disk access, etc.

Running time

- First, assume that the number of operations performed by A on input size n is dependent only on n, not the values of the data
  - If $f(n)$ is $\Theta(g(n))$, we say that A **is** $\Theta(g(n))$ or that A **is of order** $g(n)$
- If the number of operations depends not only on n but also on the values of the data
  - Let $b(n)$ be the **minimum** number of operations where $b(n)$ is $\Theta(g(n))$, then we say that **in the best case**, A **is** $\Theta(g(n))$ or that A **has a best case order of** $g(n)$
  - Let $w(n)$ be the **maximum** number of operations where $w(n)$ is $\Theta(g(n))$, then we say that **in the worst case**, A **is** $\Theta(g(n))$ or that **A has a worst case order of** $g(n)$

Time

Approximate Time for $f(n)$ Operations Assuming One Operation Per Nanosecond				
$f(n)$	$n = 10$	$n = 1{,}000$	$n = 100{,}000$	$n = 10{,}000{,}000$
$\log_2 n$	3.3×10^{-9} s	10^{-8} s	1.7×10^{-8} s	2.3×10^{-8} s
n	10^{-8} s	10^{-6} s	0.0001 s	0.01 s
$n \log_2 n$	3.3×10^{-8} s	10^{-5} s	0.0017 s	0.23 s
n^2	10^{-7} s	0.001 s	10 s	27.8 hours
n^3	10^{-6} s	1 s	11.6 days	31,668 years
2^n	10^{-6} s	3.4×10^{284} years	3.1×10^{30086} years	2.9×10^{3010283} years

Computing running time

- With a single `for` (or other) loop, we simply count the number of operations that must be performed:

```
int p = 0;
int x = 2;
for (int i = 2; i <= n; ++i) {
     p = (p + i)*x;
}
```

- Counting multiplies and adds, $n - 1$ iterations times 2 operations $= 2n - 2$
- As a polynomial, $2n - 2$ is $\Theta(n)$

Nested loops

- When loops do not depend on each other, we can simply multiply their iterations (and asymptotic bounds)

```
int p = 0;
for (int i = 2; i <= n; ++i) {
     for (int j = 3; j <= n; ++j) {
          ++p;
     }
}
```

- Clearly $(n - 1)(n - 2)$ is $\Theta(n^2)$

Trickier nested loops

- When loops depend on each other, we have to do more analysis
  What's the running time here?

```
int s = 0;
for (int i = 1; i <= n; ++i) {
     for (int j = 1; j <= i; ++j) {
          s = s + j*(i - j + 1);
     }
}
```

- Arithmetic sequence saves the day

Iterations with floor

- When loops depend on floor, what happens to the running time?

```
int a = 0;
for (int i = n/2; i <= n; ++i) {
     a = n - i;
}
```

- Floor is used implicitly here, because we're using integer division
- What's the running time? Hint: Consider n as odd or as even separately

Upcoming

Next time…

- Finish algorithmic efficiency
- Exponential and logarithmic functions

Reminders

- **Start Assignment 1**
- Read 11.4
  - Prepare a three-sentence summary
- **Office hours are canceled today**