Week 4 - Friday

COMP 2230

Last time

- Examples of SoLHRRwCC's
- General recursive definitions

Questions?

Assignment 2

Review

Growth of Functions

Growth of functions

- Power functions:
  - $p_a(x) = x^a$ where $a, x \geq 0$
- Increasing and decreasing functions
- 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 $O(g(x))$, 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 $x > k$

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 $\Theta(x^n)$

Exponential and logarithmic orders

- For all real numbers b and r with $b > 1$ and $r > 0$
  - $\log_b x \leq x^r$, for sufficiently large x
  - $x^r \leq b^x$, for sufficiently large x
- These statements are equivalent to saying for all real numbers b and r with $b > 1$ and $r > 0$
  - $\log_b x$ is $O(x^r)$
  - x^r is $O(b^x)$

Running time for algorithms

- 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** $g(n)$

Computing running time

- With a single `for` (or other) loop, we simply count the number of operations that must be performed
- When loops do not depend on each other, we can simply multiply their iterations (and asymptotic bounds)
- When loops depend on each other, we should analyze the loops like a series
- Repeated divisions by a constant k **often** lead to a $\log_k n$ term

Far too easy

- What's the Big Theta bound if n is the length of **`values`**?

```
public static double average(int[] values) {
    double sum = 0.0;
    for(int i = 0; i < values.length; ++i) {
        sum += values[i];
    }
    return sum / values.length;
}
```

A little tougher

- What's the Big Theta bound if n is the length of **`people`**?

```
public static void meeting(Person[] people) {
    for(int i = 0; i < people.length; ++i) {
        for(int j = i + 1; j < people.length; ++j) {
            people[i].shakeHands(people[j]);
        }
    }
}
```

Watch your step

- What's the Big Theta bound if n is **n**?

```
public static void powers(int n) {
    int i = 1;
    while(i <= n) {
        System.out.print(i);
        i *= 2;
    }
}
```

Keep watching your step

- What's the Big Theta bound if n is **n**?

```
for(int i = 1; i < n; ++i) {
    for(int j = 0; j < n; j += i) {
        System.out.println("*");
    }
}
```

Steps on steps

- What's the Big Theta bound if n is **n**?

```
for(int i = 1; i < n; i *= 2) {
    for(int j = 1; j < n; j *= 3) {
        System.out.println("#");
    }
}
```

Steps on steps on steps

- What's the Big Theta bound if n is **n**?

```
int counter = 1;
for(int i = 1; i <= n; ++i) {
    for(int j = 0; j < counter; ++j) {
        System.out.println("$");
    }
    counter *= 2;
}
```

Switch it up

- What's the Big Theta bound if n is **n**?

```
int counter = 1;
for(int i = 1; i <= n; ++i) {
    for(int j = 0; j < n/counter; ++j) {
        System.out.println("$");
    }
    counter *= 2;
}
```

Back to your roots

- What's the Big Theta bound if n is **n**?

```
for(int i = 0; i*i < n; ++i) {
    for(int j = 0; j < n; ++j) {
        System.out.println("%");
    }
}
```

Sequences and Induction

Sequences

- Mathematical sequences can be represented in **expanded form** or with **explicit formulas**
- Examples:
  - $2, 5, 10, 17, 26, \ldots$
  - $a_i = i^2 + 1, \quad i \geq 1$
- **Summation notation** is used to describe a summation of some part of a sequence

$$\sum_{k=m}^{n} a_k = a_m + a_{m+1} + a_{m+2} + \ldots + a_n$$

- **Product notation** is used to describe a product of some part of a sequence

$$\prod_{k=m}^{n} a_k = a_m \cdot a_{m+1} \cdot a_{m+2} \cdot \ldots \cdot a_n$$

Simplify

- Simplify the following summation to a closed-form expression

$$\sum_{i=5}^{n+2} 3i + n$$

Proof by mathematical induction

- To prove a statement of the following form:
  - $\forall n \in \mathbb{Z}$, where $n \geq a$, property $P(n)$ is true
- Use the following steps:

1. Basis Step: Show that the property is true for $P(a)$
2. Induction Step:
   - Suppose that the property is true for some $n = k$, where $k \in \mathbb{Z}, k \geq a$
   - Now, show that, with that assumption, the property is also true for $k + 1$

Mathematical induction example

- Prove that, for all integers $n \geq 1$,

$$\frac{\sum_{i=1}^{n} 2i - 1}{\sum_{i=1}^{n} 2n + 2i - 1} = \frac{1}{3}$$

Recursion

- Using recursive definitions to generate sequences
- Writing a recursive definition based on a sequence
- Using mathematical induction to show that a recursive definition and an explicit definition are equivalent

Employing outside formulas

- Sure, intelligent pattern matching gets you a long way
- However, it is sometimes necessary to substitute in some known formula to simplify a series of terms
- Recall
  - Geometric series: $1 + r + r^2 + \cdots + r^n = \frac{r^{n+1}-1}{r-1}$
  - Arithmetic series: $1 + 2 + 3 + \ldots + n = \frac{n(n+1)}{2}$

Changing from iteration to recursion

- Consider the following formula:
  - $s_i = 5^i$, for $i \geq 0$

- Convert it into a recursive formula:
  - State the recurrence relation
  - Define a starting term (s_0)

Recursive sequence example

- $g_k = \frac{g_{k-1}}{g_{k-1}+2}$ for all integers $k \geq 1$
- g_0 = 1

- Give an explicit formula for this recurrence relation
- **Hint:** Use the method of iteration

Solving second order linear homogeneous recurrence relations with constant coefficients

- To solve sequence $a_k = Aa_{k-1} + Ba_{k-2}$
- Find its characteristic equation $t^2 - At - B = 0$
- If the equation has two distinct roots r and s
  - Substitute a_0 and a_1 into $a_n = Cr^n + Ds^n$ to find C and D
- If the equation has a single root r
  - Substitute a_0 and a_1 into $a_n = Cr^n + Dnr^n$ to find C and D
- **There will be one of these on the exam**

Ticket Out the Door

Upcoming

Next time…

- Exam 1!

Reminders

- Finish Assignment 2
  - Due tonight by midnight!
- Study for Exam 1
  - Next Monday!