Announcements

- Quiz 4 is today

- Homework 5 is due on Thursday

- Read 3.2 (Growth of Functions) and 5.4-5.5 (Recursive Algorithms and Program correctness)

- Pick up your exam 1 now
  - Solutions posted online

- Kunyao’s office hours are moved to 4-6pm today from 2:30-4:30.
Exam 1 Grades (17.5% of your final grade)

Average = 137.4, Median = 143
Exam 1 Grades  (17.5% of your final grade)

Average = 91.6%, Median = 95.3%
Extra Examples

1) Using the Principle of Mathematical Induction, prove that for all positive integers \( n \):
\[
1 + 3 + 5 + \ldots + (2n - 1) = n^2.
\]
(Solution) Extra Example #1

Using the Principle of Mathematical Induction, prove that for all positive integers \( n \):
\[
1 + 3 + 5 + \ldots + (2n - 1) = n^2.
\]

**Base Step:** \( P(1) \) is true since \( 1 = 1^2 \).

**Inductive Step:** \( P(k) \) \( \rightarrow \) \( P(k + 1) \):
\[
1 + 3 + \ldots + (2k + 1) = k^2 + (2k + 1) = (k + 1)^2.
\]
Extra Examples

2) Using the Principle of Mathematical Induction to prove that

\[ 1 \times 1! + 2 \times 2! + 3 \times 3! + \ldots + n \times n! = (n + 1)! - 1 \]

for all \( n \geq 1 \).
(Solution) Extra Example #2

Using the Principle of Mathematical Induction to prove that \(1 \times 1! + 2 \times 2! + 3 \times 3! + \ldots + n \times n! = (n + 1)! - 1\) for all \(n \geq 1\).

Base Step: \(P(1)\) is true since \(1 \times 1! = 1\) and \(2! - 1 = 1\).

Inductive Step: \(P(k) \rightarrow P(k + 1)\):

\[
\begin{align*}
1 \cdot 1! + 2 \cdot 2! + \ldots + (k + 1)(k + 1)!
&= (k +1)!-1+ (k +1)(k +1)!
&= (k +1)![1+ (k +1)]-1
&= (k +1)!(k + 2) -1
&= (k + 2)!-1
\end{align*}
\]
Extra Examples

3) Use mathematical induction to prove that every amount of postage of six cents or more can be formed using 3-cent and 4-cent stamps.
(Solutions) Extra Example #3

Use mathematical induction to prove that every amount of postage of six cents or more can be formed using 3-cent and 4-cent stamps.

- **Base Step**: $P(6)$ : Six cents postage can be made from two 3-cent stamps.

- **Inductive Step**: $P(k) \rightarrow P(k + 1)$:
  - Either replace a 3-cent stamp by a 4-cent stamp or else (if there are only 4-cent stamps in the pile of stamps making $k$ cents postage) replace two 4-cent stamps by three 3-cent stamps.
Growth of Functions

- Let $f$ and $g$ be functions from the set of integers or the set of real numbers to the set of real numbers. We say $f(x) = O(g(x))$ if there are constant $C$ and $k$ s.t.
  - $f(x) \leq Cg(x)$ for all $x > k$

- Intuitively, $f(x)$ grows more slowly in $x$ than $g(x)$
- Called Big Oh notation
  - Sometimes we say $f(x) = O(g(x))$ or $f(x)$ is $O(g(x))$
Growth of Functions

- Example: \( f(x) = x^2 + 2x + 1 \) is \( O(x^2) \)

- \( f(x) = x^2 + 2x + 1 \leq x^2 + 2x^2 + x^2 = 4x^2 \) --true when \( x > 1 \)

- Choose \( C = 4 \) and \( k = 1 \)

- Example \( n! = O(n^n) \)
- \( n! = (1)(2)(3)(4)\ldots(n-1)(n) \leq (n)(n)(n)\ldots = n^n \)
Growth of Functions

- If \( f(x) \) is \( O(g(x)) \) and \( f'(x) \) is \( O(g'(x)) \) then \( (f(x)+f'(x)) \) is \( O(\max\{g(x),g'(x)\}) \)

- \( f(x) + f'(x) \leq Cg(x) + Cg'(x) \) for sufficiently large \( x \) and large \( C \)

- \( Cg(x) + Cg'(x) \leq 2C \max\{g(x),g'(x)\} \)
Growth of Functions

• Let $f$ and $g$ be functions from the set of integers or the set of real numbers to the set of real numbers. We say $f(x) = \Omega(g(x))$ if there are constant $C$ and $k$ s.t.
  – $f(x) \geq Cg(x)$ for all $x > k$

• Intuitively, $f(x)$ grows faster in $x$ than $g(x)$

• Called Big Omega notation
Growth of Functions

- Example: $f(x) = x^2 + 2x + 1$ is $\Omega(x^2)$
  - $f(x) = x^2 + 2x + 1 \geq x^2$ --true when $x > 1$
  - Choose $C = 1$ and $k = 1$
  - Note it is also $\Omega(x)$ or $\Omega(1)$
Growth of Functions

- Let $f$ and $g$ be functions from the set of integers or the set of real numbers to the set of real numbers. We say $f(x) = \Theta(g(x))$ if
  - $f(x)$ is $O(g(x))$ and $f(x)$ is $\Omega(f(x))$
Growth of Functions

• Let \( f(x) = x^3 + 100x + 100,000 \) and \( g(x) = x^3 \)
• Show that \( f(x) \) is \( \Theta(g(x)) \)

• \( f(x) \) is \( O(g(x)) \)
  – \( C= 1, k=1 \)
  – \( x^3 + 100x + 100,000 > x^3 \)

• \( f(x) = \Omega(g(x)) \)
  – \( C = 100, k=100,000 \)
  – \( 100(x^3) > x^3 + 100x + 100,000 \) when \( x > 100,000 \)
Growth of Functions

- Let \( f(x) = x^3 + 100x + 100,000 \) and \( g(x) = x^3 \)
- Show that \( f(x) \) is \( \Theta(g(x)) \)

- \( f(x) \) is \( O(g(x)) \)
  - \( C = 1, k=1 \)
  - \( x^3 + 100x + 100,000 > x^3 \)

- \( f(x) = \Omega(g(x)) \)
  - \( C = 100, k=100,000 \)
  - \( 100(x^3) > x^3 + 100x + 100,000 \) when \( x > 100,000 \)
Growth of Functions

• Let $f$ and $g$ be functions from the set of integers or the set of real numbers to the set of real numbers. We say $f(x) = o(g(x))$ if
  
  – For every positive constant $c$ there exists a constant $N$
  – $f(x) \leq c \cdot g(x)$ for all $x > N$

– Intuitively, $f(x)$ grows more slowly in $x$ than $g(x)$, but also $f(x)$ is not $\Omega(g(x))$

– Called Little Oh notation
  • Less commonly used in CS
Growth of Functions

• Let $f$ and $g$ be functions from the set of integers or the set of real numbers to the set of real numbers. We say $f(x) = \omega(g(x))$ if
  
  – For every positive constant $c$ there exists a constant $N$
  – $f(x) > c \cdot g(x)$ for all $x > N$

  – Intuitively, $f(x)$ faster in $x$ than $g(x)$, but also $f(x)$ is not $O(g(x))$

  – Called Little Omega notation
    • Less commonly used in CS
Growth of Functions: Why do we care?

- **Program running time**
  - We want to know how long our programs will run when the input is large
  - If the input to a program has size $n$, typically want to know if the program is $O(f(n))$ for the smallest function $f(n)$
  - Typically the best indicator of how fast the program is

- **Downside**
  - Constant hidden in the big oh can be large

- **A=1**
- **For (i=1 to n)**
  - A++

- **How long does this program take?**
  - $O(n)$
Growth of Functions: Why do we care?

• A=1
• For (i=1 to n)
  – For (j=1 to n)
    • A++

• How long does this program take?
  – O(n^2)
Growth of Functions: Why do we care?

- A=1, B=2
- For (i=1 to n)
  - For (j=1 to n)
    - A=A+1
    - B=B+1
    - A= A+B

- How long does this program take?
  - O(n^2)
Useful Example

- $f(n) = \log(n!)$ is $O(n \log n)$
  - $n! \leq n^n$
  - Therefore, $\log(n!) \leq \log(n^n)$
  - $\log(n^n) = n \log n$
Quiz 4