Today's Outline

- **Admin**: Assignment #1 due next thurs. at 11:59pm
- Asymptotic analysis

---

Today's Outline

- **Admin**: Assignment #1 due next thurs. at 11:59pm
- Asymptotic analysis

---

**Asymptotic Analysis**

CSE 373
Data Structures & Algorithms
Ruth Anderson
Spring 2007

---

**Linear Search vs Binary Search**

<table>
<thead>
<tr>
<th></th>
<th>Linear Search</th>
<th>Binary Search</th>
</tr>
</thead>
<tbody>
<tr>
<td>Best Case</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Worst Case</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Fast Computer vs. Slow Computer**

---

**Fast Computer vs. Smart Programmer (round 1)**

---
Fast Computer vs. Smart Programmer (round 2)

Asymptotic Analysis
• Asymptotic analysis looks at the order of the running time of the algorithm
  – A valuable tool when the input gets “large”
  – Ignores the effects of different machines or different implementations of the same algorithm
• Intuitively, to find the asymptotic runtime, throw away the constants and low-order terms
  – Linear search is $T(n) = 3n + 2 \in O(n)$
  – Binary search is $T(n) = 4 \log_2 n + 4 \in O(\log n)$

Remember: the fastest algorithm has the slowest growing function for its runtime

Asymptotic Analysis
• Eliminate low order terms
  – $4n + 5 \Rightarrow 0$
  – $0.5 n \log n + 2n + 7 \Rightarrow 0$
  – $n + 2^n + 3n \Rightarrow 0$
• Eliminate coefficients
  – $4n \Rightarrow 0$
  – $0.5 n \log n \Rightarrow 0$
  – $n \log n \Rightarrow 0$

Order Notation: Intuition
Although not yet apparent, as $n$ gets “sufficiently large”, $f(n)$ will be “greater than or equal to” $g(n)$

Definition of Order Notation
• Upper bound: $T(n) = O(f(n))$ Big-O
  Exist constants $c$ and $n_0$ such that
  $T(n) \leq c f(n)$ for all $n \geq n_0$
• Lower bound: $T(n) = \Omega(g(n))$ Omega
  Exist constants $c$ and $n_0$ such that
  $T(n) \geq c g(n)$ for all $n \geq n_0$
• Tight bound: $T(n) = \Theta(f(n))$ Theta
  When both hold:
  $T(n) = O(f(n))$
  $T(n) = \Omega(f(n))$

Order Notation: Definition
$O(f(n))$: a set or class of functions
$g(n) \in O(f(n))$ iff there exist constants $c$ and $n_0$ such that:
$g(n) \leq c f(n)$ for all $n \geq n_0$

Example: $g(n) = 1000n$ vs. $f(n) = n^2$
Is $g(n) \in O(f(n))$?
Pick: $n_0 = 1000$, $c = 1$
**Notation Notes**

**Note:** Sometimes, you’ll see the notation:

\[ g(n) = O(f(n)). \]

This is equivalent to:

\[ g(n) \in O(f(n)). \]

**However:** The notation

\[ O(f(n)) = g(n) \]

is meaningless!

(in other words big-O is not symmetric)

---

**Big-O: Common Names**

- constant: \( O(1) \)
- logarithmic: \( O(\log n) \) (\( \log_k n, \log n^2 \in O(\log n) \))
- linear: \( O(n) \)
- log-linear: \( O(n \log n) \)
- quadratic: \( O(n^2) \)
- cubic: \( O(n^3) \)
- polynomial: \( O(n^k) \) (\( k \) is a constant)
- exponential: \( O(c^n) \) (\( c \) is a constant > 1)

---

**Meet the Family**

- \( O(f(n)) \) is the set of all functions asymptotically less than or equal to \( f(n) \)
- \( o(f(n)) \) is the set of all functions asymptotically strictly less than \( f(n) \)
- \( \Omega(f(n)) \) is the set of all functions asymptotically greater than or equal to \( f(n) \)
- \( \omega(f(n)) \) is the set of all functions asymptotically strictly greater than \( f(n) \)
- \( \Theta(f(n)) \) is the set of all functions asymptotically equal to \( f(n) \)

---

**Meet the Family, Formally**

- \( g(n) \in O(f(n)) \) iff
  
  There exist \( c \) and \( n_0 \) such that \( g(n) \leq c f(n) \) for all \( n \geq n_0 \)
  
  - \( g(n) \in o(f(n)) \) iff
    
    There exists a \( n_0 \) such that \( g(n) < c f(n) \) for all \( c \) and \( n \geq n_0 \)
  
  - \( g(n) \in \Omega(f(n)) \) iff
    
    There exist \( c \) and \( n_0 \) such that \( g(n) \geq c f(n) \) for all \( n \geq n_0 \)
  
  - \( g(n) \in \omega(f(n)) \) iff
    
    There exists a \( n_0 \) such that \( g(n) > c f(n) \) for all \( c \) and \( n \geq n_0 \)

---

**Big-Omega et al. Intuitively**

\[ \begin{array}{ccc}
\text{Asymptotic Notation} & \text{Mathematics Relation} \\
O & \leq \\
\Omega & \geq \\
\theta & =
\end{array} \]
Pros and Cons of Asymptotic Analysis

Types of Analysis
Two orthogonal axes:

- bound flavor
  - upper bound (O, o)
  - lower bound (Ω, ω)
  - asymptotically tight (θ)

- analysis case
  - worst case (adversary)
  - average case
  - best case
  - "amortized"

Algorithm Analysis Examples
- Consider the following program segment:
  ```
  x := 0;
  for i = 1 to N do
    for j = 1 to i do
      x := x + 1;
  ```
- What is the value of x at the end?

Analyzing the Loop
- Total number of times x is incremented is executed =
  \[1 + 2 + 3 + \ldots + \sum_{i=1}^{N} i = \frac{N(N+1)}{2}\]
- Congratulations - You’ve just analyzed your first program!
  - Running time of the program is proportional to \(N(N+1)/2\) for all N
  - Big-O ??

Which Function Grows Faster?
\(n^3 + 2n^2\) vs. \(100n^2 + 1000\)

Which Function Grows Faster?
\(n^3 + 2n^2\) vs. \(100n^2 + 1000\)