Discrete Mathematical Structures
CS 2233 Lecture Fourteen

Prof. William Winsborough
March 5, 2009

Business

- **Turn in Homework 6 today**
- **Recall:**
  - **Homework 7**, due Thursday 3/19
    - 3.2: 2, 4, 6, 8, 14, 20, 22
    - 3.3: 2
  - Read Sections 3.2 and 3.3
- Questions???

Example

- Let $f(x) = x^2 + 2x + 1$ and $g(x) = x^2$
- $f(x)$ is $O(g(x))$
  - k = 1 and C = 4 (witnesses)
  - k = 2 and C = 3
- $g(x)$ is $O(f(x))$
  - k = 1 and C = 1

More Examples

- Sum of first n positive integers
  $1+2+\ldots+n \le \underbrace{n+n+\ldots+n}_{n} = n^2$
  So $\sum_{i=1}^{n} i$ is $O(n^2)$
- It follows that the complexity of both bubble sort and insertion sort is $O(n^2)$

Still More Examples

- $f(n) = n!$ is the product of the first n positive integers
  $n! = \underbrace{1\cdot 2\cdot \ldots \cdot n}_{n} \le n\cdot n\cdot \ldots \cdot n = n^n$

  So $n!$ is $O(n^n)$

Big-O and Sums

- Def: If f_1, f_2 are real-valued functions, (f_1+f_2) is the function such that that $(f_1+f_2)(x) = f_1(x)+f_2(x)$ for all x
- Theorem:
  If $f_1(x)$ is $O(g_1(x))$ and $f_2(x)$ is $O(g_2(x))$, then $(f_1+f_2)(x)$ is $O(\max(|g_1(x)|, |g_2(x)|)$
- Corollary:
  If $f_1(x)$ and $f_2(x)$ are each $O(g(x))$, then $(f_1+f_2)(x)$ is $O(g(x))$

Big-O and Products

- Def: If f_1, f_2 are real-valued functions, (f_1f_2) is the function such that that $(f_1f_2)(x) = f_1(x) \cdot f_2(x)$ for all x
- Theorem:
  If $f_1(x)$ is $O(g_1(x))$ and $f_2(x)$ is $O(g_2(x))$, then $(f_1f_2)(x)$ is $O(g_1(x)\, g_2(x))$

Example

- Claim: $\sum_{i=1}^{n} i$ is $\Theta(n^2)$
  - We have already shown it is $O(n^2)$,
  - We have to show it is $\Omega(n^2)$
    - Recall that $\sum_{i=1}^{n} i = n(n+1)/2$
    - Taking C = ½ and k = 0, we have $n^2/2 + n/2 \geq Cn^2$ for all n>k

Significance of Theorems on Sums and Products

- A polynomial of degree k is $O(x^k)$

Big-Omega and Big-Theta

- Big-O provides an upper bound on function growth
- Big-Omega gives a lower bound
  - Def: f(x) is $\Omega(g(x))$ if there are positive constants C and k such that $|f(x)| \geq C|g(x)|$ whenever x>k
- Big-Theta gives both
  - Def: f(x) is $\Theta(g(x))$ if f(x) is O(g(x)) and f(x) is $\Omega(g(x))$
  - In this case we say f(x) is *of order* g(x)

Computational Complexity of Algorithms

- This is where Big-O, Big-Omega, and Big-Theta get used
- Complexity is measured principally in terms of two resources
  - Time Complexity
  - Space Complexity
    - Discussed more in course on data structures
- Worst-case complexity vs. Average-case

Worst- versus Average-case Complexity

- Possible inputs are partitioned into cases that elicit the same behavior (*input cases*)
- *Worst-case complexity* analyzes the input case that maximizes cost of executing the algorithm
- *Average-case complexity* considers all input cases
  - The *average-case complexity* of executing an algorithm is:
    - The sum over all input cases of the cost of each case times its probability
    - Can be difficult to calculate
    - Assumes you know the probability of each input case
  - Also called *expected-case complexity* or, less formally, *expected cost*