CS 2022 : DATA STRUCTURES & ALGORITHMS

Lecture 2: Complexity Analysis

Malaka Walpola

Outline

- Analyzing Algorithms
- Analysis of Insertion Sort
- Analysis of Bubble Sort
- Asymptotic Notation

Learning Outcomes

- After successfully studying contents covered in this lecture, students should be able to,
  - explain the major factors considered for analyzing algorithms
  - explain the use of asymptotic analysis to examine the growth of functions
  - express the time complexity of simple (non recursive) algorithms using asymptotic notation

Analyzing Algorithms

- What are the factors that affect the running time of a program?
  - The processing
  - CPU speed
  - Memory
  - Size of input data set
  - Nature of input data set

Analyzing Algorithms

- Factors to Consider
  - Speed/Amount of work done/Efficiency
  - Space efficiency/ Amount of memory used
  - Simplicity
- We want an analysis that does not depend on
  - CPU speed
  - Memory

Analyzing Algorithms

- Why?
  - We want to have **a hardware independent** analysis
- Thus, we use Asymptotic Analysis
  - Ignore machine dependant constants
  - Look at the growth of the running time
    - Running time of an algorithm as a function of input size ***n*** **for large** ***n***
  - Written using **Asymptotic Notation**

Analyzing Algorithms

- Kinds of Analysis
  - Worst-case: (usually)
  - Average-case: (sometimes)
    - Need assumption of statistical distribution of inputs.
  - Best-case: (bogus?)

Analysis of Insertion Sort

Code	Cost	Times
INSERTION-SORT(A)	----	---
1. for j = 2 to A.length	C_1	n
2. key = A[j]	C_2	n-1
3. //…………	0	n-1
4. i = j-1	C_4	n-1
5. while i > 0 and A[i] > key	C_5	$\sum_{j=2}^{n} t_j$
6. A[i+1] = A[i]	C_6	$\sum_{j=2}^{n} (t_j - 1)$
7. i = i-1	C_7	$\sum_{j=2}^{n} (t_j - 1)$
8. A[i+1] = key	C_8	n-1

- n = A.length
- t_j = the number of times the while loop test in line 5 is executed

Analysis of Insertion Sort

- $$T(n) = c_1 n + c_2(n-1) + c_4(n-1) + c_5 \sum_{j=2}^{n} t_j$$

$$+ c_6 \sum_{j=2}^{n} \left(t_j - 1\right) + c_7 \sum_{j=2}^{n} \left(t_j - 1\right) + c_8(n-1)$$

- Best Case – Sorted Array
  - t_j = 1 for j=2, 3, ……, n

Analysis of Insertion Sort

- $$T(n) = c_1 n + c_2(n-1) + c_4(n-1) + c_5\sum_{j=2}^{n} t_j + c_6\sum_{j=2}^{n}\left(t_j - 1\right) + c_7\sum_{j=2}^{n}\left(t_j - 1\right) + c_8(n-1)$$

- Worst Case – Reverse Sorted Array
  - t_j = j for j=2, 3, ……, n
  - $\sum_{j=2}^{n} j = \frac{n(n+1)}{2} - 1$ & $\sum_{j=2}^{n} (j-1) = \frac{n(n-1)}{2} - 1$

Analysis of Bubble Sort

Code	Cost	Times
BUBBLE-SORT(A)		
1. do		
2. swapped ← false		
3. for i = 2 to A.length		
4. if A[i-1] > A[i]		
5. temp ← A[i]		
6. A[i] ← A[i-1]		
7. A[i-1] ← temp		
8. swapped ← true		
9. while swapped		

Asymptotic Notation

- Θ-Notation
  - Asymptotically tight bound
  - $\Theta(g(n)) = \{f(n)$: there exist positive constants c_1, c_2, and n_0 such that $0 \le c_1 g(n) \le f(n) \le c_2 g(n)$ for all $n \ge n_0\}$

Asymptotic Notation

- O-Notation
  - Asymptotic upper bound
  - $O(g(n)) = \{f(n)$: there exist positive constants c, and n_0 such that $0 \leq f(n) \leq cg(n)$ for all $n \geq n_0\}$
- A General Guideline
  - Ignore lower order terms
  - Ignore leading constants

The O Notation - Examples

f(x) = 3x

g(x) = x

C = 3, N =0

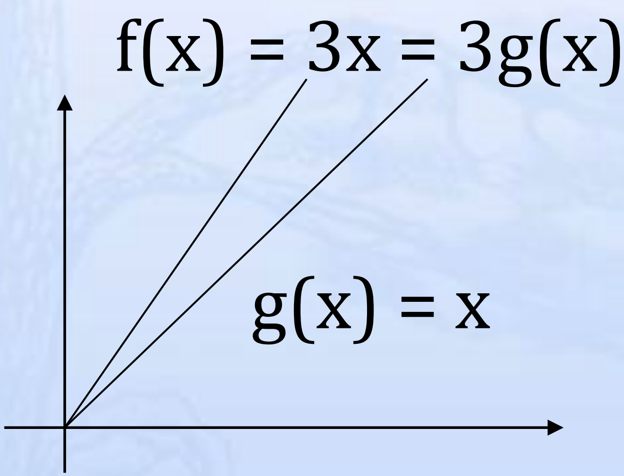
*f(x) ≤ 3*g(x)*

f(x) = 4+7x

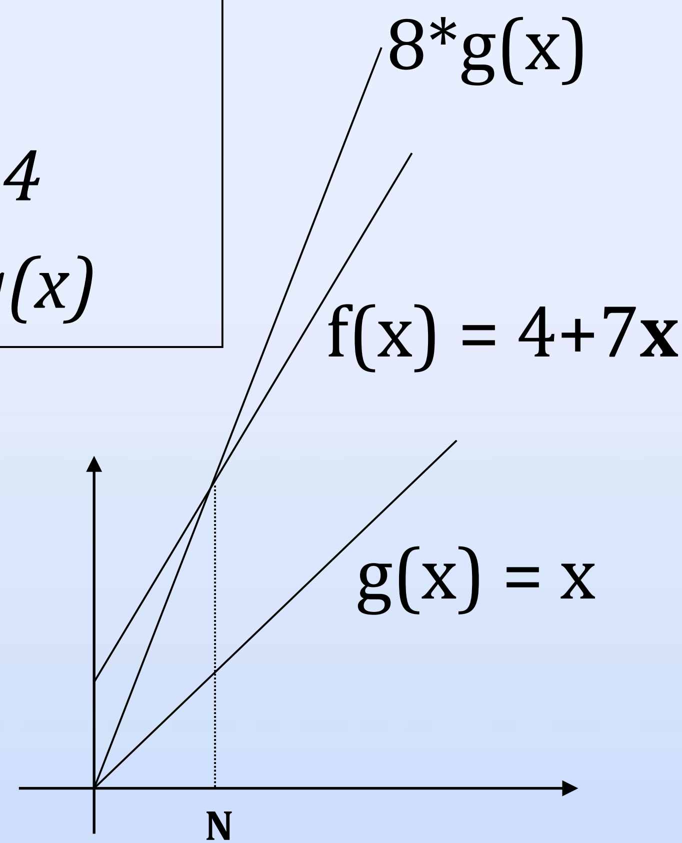
g(x) = x

C = 8, N =4

*f(x) ≤ 8*g(x)*



$3x = O(x)$



$4+7x = O(x)$

The Θ Notation - Examples

$f(x) = 3x$

$g(x) = x$

$C_1 = 1, N = 0, C_2 = 3$

Show

$g(x) \le f(x) \le 3*g(x)$

$g(x) = x < f(x)$

$f(x) = 3x = 3g(x)$

Therefore

$f(x) = 3x = \theta(x) = \theta(g(x))$

$f(x) = 4+7x$

$g(x) = x$

$C_1 = 7, N = 4, C_2 = 8$

Show

$7*g(x) \le f(x) \le 8*g(x)$

$7*g(x) = 7x < f(x)$

$f(x) = 7x+4 \le 8x$ (when $x \ge 4$)

Therefore

$f(x) = 7x+4 = \theta(x) = \theta(g(x))$

The O Notation - Exercises

$f(x) = x+100x^2$

$g(x) = x^2$

Show that

$f(x) = O(g(x))$

$f(x) = 20+x+5x^2+8x^3$

$g(x) = x^3$

Show that

$f(x) = O(g(x))$

The Θ Notation - Exercises

$f(x) = x+100x^2$

$g(x) = x^2$

Show that

$f(x) = \theta(g(x))$

$f(x) = 20+x+5x^2+8x^3$

$g(x) = x^3$

Show that

$f(x) = \theta(g(x))$

Asymptotic Notation

- Example

$$T(n) = c_1 n + c_2(n-1) + c_4(n-1) + c_5 \sum_{j=2}^{n} t_j$$

$$+ c_6 \sum_{j=2}^{n} \left(t_j - 1\right) + c_7 \sum_{j=2}^{n} \left(t_j - 1\right) + c_8(n-1)$$

- Macro Substitution
  - Convention: A set in a formula represents an anonymous function in the set
  - $f(n) = n^3 + O(n^2)$
    - $f(n) = n^3 + h(n)$ for some $h(n) \in O(n^2)$

Asymptotic Notation

- O-Notation
  - May or may not be asymptotically tight
- Other Notations (Self-Study)
  - Ω-notation
    - Asymptotic lower bound
    - May or may not be asymptotically tight
  - o-notation
    - Asymptotic upper bound (Not asymptotically tight)
  - ω-notation
    - Asymptotic lower bound (Not asymptotically tight)

HOMEWORK

- Study the Ω, o and ω asymptotic notations
  - What are the relationships between the Θ, O, Ω, o and ω notations?
- Study the optimized Bubble sort algorithms given in the slides
  - Analyze the optimized versions of Bubble sort for worst case time complexity
  - Is there a difference in worst case time complexity?
  - Is there an easy method to analyze only the worst case time complexity?

Optimizing Bubble Sort (1)

```
OPTIMIZED-BUBBLE-SORT(A)
 for j = A.length to 2
   swapped = false
   for i = 2 to j
     if A[i-1] > A[i]
       temp = A[i]
       A[i] = A[i-1]
       A[i-1] = temp
       swapped = true
   if (!swapped)
     break;
```

Optimizing Bubble Sort (2)

```
OPTIMIZED-BUBBLE-SORT(A)
n = A.length
do
   swapped = false
     for i = 2 to n
       if A[i-1] > A[i]
          temp = A[i]
          A[i] = A[i-1]
          A[i-1] = temp
          swapped = true
          newLimit = i-1
     n = newLimit
while swapped
```

REFERENCES

[1] T.H. Cormen, C.E. Leiserson, R.L. Rivest and C. Stein, *Introduction to Algorithms*, 3rd Ed. Cambridge, MA, MIT Press, 2009.

[2] S. Baase and Allen Van Gelder, *Computer Algorithms: Introduction to Design and Analysis*, 3rd Ed. Delhi, India, Pearson Education, 2000.

[3] Wikipedia , “Insertion sort”, Oct. 09, 2012. [Online]. http://en.wikipedia.org/wiki/Insertion_sort. [Accessed: Oct. 22, 2012].

[4] Wikipedia , “Bubble sort”, Oct. 17, 2012. [Online]. http://en.wikipedia.org/wiki/Bubble_sort. [Accessed: Oct. 22, 2012].

[5] Lecture slides from Prof. Erik Demaine of MIT, available at http://dspace.mit.edu/bitstream/handle/1721.1/37150/6-046JFall-2004/NR/rdonlyres/Electrical-Engineering-and-Computer-Science/6-046JFall-2004/B3727FC3-625D-4FE3-A422-56F7F07E9787/0/lecture_01.pdf

Reading Assignment

- What we covered today
  - IA –Sections 3.1
  - Topic-wise
    - Asymptotic Notation: Section 3.1 of IA
- Next class
  - Recursion and Divide & Conquer Approaches and Merge Sort: Section 2.3.1