## Design and Analysis of Algorithms

## CS375 Spring 2016

## Theory Assignment 1

Release Date: 2/3/2016

Due: 2/10/2016 (Wednesday) at start of class

Remember to include the following statement at the start of your answers with a signature by the side.

“I have done this assignment completely on my own. I have not copied it, nor have I given my solution to anyone else. I understand that if I am involved in plagiarism or cheating I will have to sign an official form that I have cheated and that this form will be stored in my official university record. I also understand that I will receive a grade of **0** for the involved assignment for my first offense and that I will receive a grade of **“F” for the course** for any additional offense.” (Keni Mou)



All solutions of theory assignments must be typed (no handwritten solutions) and submitted in hard copy. Advance electronic submission to the TA is acceptable if the student is expected to miss the class on the due date.

1. (10 points) Given the pseudo code below for bubble sort:

BubbleSort(A)

**Line1: for** i = 1 **to** (length[A]-1)

//store next smallest element in A[i]

**Line2:** **for** j = length[A] **downto** (i + 1)

**Line3:** **if** A[j] < A[j-1]

**Line4:** swap A[j] and A[j-1];

a) (5 points) Let length[A] = *n*. What is the count for BubbleSort(A)? Show the steps necessary to derive your final answer. This question requires you to use the instruction count method from the textbook (also introduced in lecture 2 slides). **Answers using asymptotic notations will receive 0 point.**

|  |  |  |
| --- | --- | --- |
| BubbleSort(A) | Cost | Times |
| **for** i = 1 **to** (length[A]-1) | c1 | n |
| **for** j = length[A] **downto** (i + 1) | c2 |  |
| **if** A[j] < A[j-1] | c3 |  |
| swap A[j] and A[j-1]; | c4 |  |

, depends on line 3.

b) (5 points) Show the worse case and best case time complexity in term of instruction counts.

Worst case: when s(i) = i-1 for all i,

Best case: s(i) = 0 for all i,

1. (28 points) Fill in all the missing values. For the *f(n)* column, you need to compute the sums and fill in the exact format of *f(n)* for the last two rows. For the last three columns, you need to fill in each cell with either yes or no.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *f(n)* | *g(n)* | *f(n)* = O(*g(n)*) | *f(n)* =Ω(*g(n)*) | *f(n)*=Θ(*g(n)*) |
| *n*2.125 | *n*2 lg *n* | No | Yes | No |
|  | *n* | Yes | No | No |
| *n*! | (*n*+1)! | Yes | No | No |
| *2n/2* | *2n* | Yes | Yes | Yes |
|  | *n*2 | Yes | Yes | Yes |
| *=* | *n4(n-1)* | Yes | No | No |

1. (10 points) Order the functions below by increasing growth rates (no justification required):

*nn*, *n*ln *n*, *n*ε (0 < ε < 1), , ln *n,* 10, *n*!, 2*n*

Let *gi*(*n*) be the *i*th function from the left after the ordering (the leftmost function has the slowest growth rate). In the order, *gi*(*n*) should satisfy *gi*(*n*) ∈ O(*gi*+1(*n*)).

10, ln n, *n*ε ,, *n*ln *n,* 2*n,n!, nn*

1. (12 points) Let *f*(*n*) and *g*(*n*) be asymptotically positive functions. Prove or show a counter example for each of the following conjectures.
   1. 

Counter example:

= = ∞

Which contradict the statement. Thus the statement is false.

* 1. .

Pf: from LHS, f(n) = O(g(n)) => there exists c, n0>0 such that 0<=f(n)<=cg(n) for all n>n0 by definition. Let c’ = 1/c, we have c’f(n) = 1/c f(n) <= 1/c \* cg(n) = g(n), which satisfies the RHS.

1. (10 points) Prove using the original definition of Θ.

Pf: Let c1 = 0.5, c2 = 1, and n0 = 10, claim:

LHS inequality:

Plug in n0 = 11,

RHS inequality: is always true.

1. (10 points) Disprove using the original definition of O.

Disproof: Assume the statement is true, there exists a constant c >0 such that for all n > n0 where n0 >0 is a constant. But n cannot be always smaller than a constant, so it contradicts the assumption.

1. (10 points) Prove *n* = (lg*n*2) using limit.

Pf: = = = ∞ => the statement holds.

1. (10 points) Prove *na* = (lgk*n*), where k>0, a>0, using limit.

Pf: = = = = = = = … = = = ∞ => the statement holds.