CSI2110 - Fall 2012 - Assignment 1

Posted: Tuesday, September 18, 2012
Due: Thursday, September 27, 2012

The assignment **must be handed in in class by Thursday, September 26, 2012**.

Exercise 1. (8 points)
Formally prove the following four statements (i.e., show a constant c and a n_0 such that ...):

1. $\sum_{i=0}^{n} 2i$ is $O(n^2)$
2. 3 is $\Theta(1)$
3. $n^2 + 4 + 100 \log n$ is $O(n^2)$
4. $\sum_{i=0}^{\log n} 2^i$ is $\Omega(n)$

Exercise 2. (2 points)
Suppose you have two algorithms A_1 and A_2 to solve a problem P. You do not know the exact running time of the two algorithms, however you know that:

- *Scenario 1.* the running time of A_1 is $O(n^2)$ and the running time of A_2 is $\Omega(n^2)$.
- *Scenario 2.* the running time of A_1 is $\Theta(n^2)$ and the running time of A_2 is $\Omega(n)$.

You would like to select the algorithm with the better running time for each scenario. Are you able to make a decision? Explain!

Exercise 3. (4 points)
Give a big-Oh characterization, in terms of n, of the running time of the following algorithms if you count **sums** as primitive operations (show your work):

Algorithm 1

Algorithm Hello(A).
Let A be an array of size n.
for $i \leftarrow 0$ **to** n **do**
    for$j \leftarrow 0$ **to** 100 **do**
        $A[i] \leftarrow A[i] + A[j]$

Algorithm 2

Algorithm Hello(B).
Let B be an array of size n.
$s \leftarrow 0$
for $i \leftarrow 1$ **to** n **do**
    for $j \leftarrow 1$ **to** i **do**
        $s = s + A[j]$

Exercise 4. (8 points)
Consider an implementation of a stack ADT using an extendable array in such a way that, when an array of size N is full, you create a new array of size $3N$. Assume you start phase 0 with an array of size 1 (i.e., phase 1 starts at the second push, which is is a special push):
1) What is the cost of the 5th push?
2) How many normal push can be performed in phase 3 ?
3) What is the cost of the i-th *special push* ?
4) What is the overall cost of performing N push (give the exact formula as well as the big-Oh characterization)?
Show your work.

Exercise 5. (4 points)
Using an auxiliary stack S we want to re-arrange the integer numbers stored in a queue Q in a way to have the even numbers dequeued before the odd numbers. Write the pseudo-code of the algorithm `ReArrange(Queue Q)` performing this re-ordering using a stack S and all the methods of the ADT Queue and the ADT Stack seen in class (enqueue, dequeue, push, pop, size etc.).