Preliminaries and Lecture 1
CS6033 Homework Assignment[*]1

Due Jan. 31th at 5:30 p.m.
Turn in this assignment as a PDF file on NYU classes
No late assignments accepted[†]

January 26, 2017

1. (5 points) Find two functions $f(n)$ and $g(n)$ such that $f(n) \in o(g(n))$ and $f(n) \notin \Theta(g(n))$. If no such functions exist, write "No such functions exist".

2. (5 points) Find two functions $f(n)$ and $g(n)$ such that $f(n) \in \Omega(g(n))$ and $f(n) \notin O(g(n))$. If no such functions exist, write "No such functions exist".

3. (10 points) In your book,[1] complete homework problem 1-1 pg 15 excluding determining the answer for $n!$ and $n \log(n)$. Assume 31 days in a month.

4. (10 points) Does the following algorithm correctly print out all the items that appear more than one time in the array (and prints no other values)? If so, prove it using a *loop invariant*! If not, prove that it does not work by providing an example on which it fails and show how this example is a counterexample.

```
PRINT_DUPLICATES($A$)
for  $i = 1$ to $A.length$
    for  $j = i$ to $A.length$
        if $A[i] == A[j]$
            PRINT($A[i]$)
```

5. (15 points) Write the pseudocode for finding all the duplicates in an array A. Your algorithm must run in time $o(n^2)$. You may call any algorithm as a subroutine that was presented in lecture 1. State what the running time is of your algorithm in big-Oh, Theta, and Omega notation.

[*] Many of these questions came from outside sources.

[†] If the class has started, your homework will be late and will not be accepted for credit.

[1] Make sure you have the third edition.

6. (20 points) Is the following algorithm to sum the items in the array $A[1..n]$ correct? If so, prove it using a *loop invariant*! If not, prove that it does not work by providing an example on which it fails and show how this example is a counterexample.

```
SUM(A)
$s = 0$
**for** $i = 1$ **to** $A.length$
    $s = s + A[i]$
**return** $(s)$
```

7. What is the run time of the following questions? Expressing your answer using big-Oh, little-oh, Theta, and Omega.

   (a) (5 points)

```
GAUSSES_SUMMATION_SERIES_1($n$)
$s = 0$
**for** $i = 1$ **to** $n$
    **for** $j = i$ to $n$
        $s = s + 1$
**return** $s$
```

   (b) (5 points)

```
GAUSSES_SUMMATION_SERIES_2($n$)
$s = 0$
**for** $i = 1$ **to** $n$
    $s = s + i$
**return** $s$
```

   (c) (5 points)

```
PRINT_REPEATED_ITEMS($A$)
**for** $i = 1$ **to** $A.length$
    **for** $j = i + 1$ **to** $A.length$
        **if** $A[i] == A[j]$
            PRINT($A[i]$)
```

8. (10 points) In your book,[2] complete homework problem 2.3-6 pg 39.

[2] Make sure you have the third edition.

9. (10 points) For the following algorithm: Show what is printed by the following algorithm[3] when called with MAXIMUM$(A, 1, 5)$ where $A = [1, 2, 3, 4, 5]$? Where the function PRINT simple prints its arguments in some appropriate manner.

```
MAXIMUM$(A, l, r)$
if $(r - l == 0)$
    return $A[r]$

$lmax =$ MAXIMUM$(A, l, \lfloor (l + r)/2 \rfloor)$
$rmax =$ MAXIMUM$(A, \lfloor (l + r)/2 \rfloor + 1, r)$
PRINT$(rmax, lmax)$
if $rmax < lmax$
    return $lmax$
else
    return $rmax$
```

10. (3 bonus points) Think of a *good*[4] exam question for the material covered in Lecture 1

[3] Of course it is a bit silly to write a divide and conquer algorithm to find the maximum item in an array. A simple loop would be much better! But this is a homework problem to test you understand how divide and conquer algorithms work.

[4] Only well thought out questions will receive the bonus points.