Assignment 1 (90 marks + 10 bonus marks)
Due Monday, Jan. 31, by 5pm.

- Assignments in MS Word format should be handed in via D2L.

---

1. (20 marks: 10 marks for algorithm design and 10 marks for implementation)

   (a) Design a recursive algorithm that determines if a string s is a palindrome, that is, it is equal to its reverse. Give the pseudo code description of your algorithm.

   (b) Implement your recursive algorithm using Java. Your program takes any string as the input, outputs "yes" if the input string is palindrome, or "no" otherwise.

2. **(total 35 marks, 5 marks per question)** Analyze the running time of the following algorithms asymptotically.

   (a) **Algorithm** *for-loop1*(n):

   $p = 1$
   for $i = 1$ to $5n^2$ do
   $\quad p = p \times i$
   return p

   (b) **Algorithm** *for-loop2*(n):

   $s = 0$
   for $i = 1$ to n do
   $\quad$ for $j = i$ to n do
   $\quad\quad s = s + i$
   return s

   (c) **Algorithm** Algorithm *WhileLoop1*(n):

   $x = 0$;
   $j = 2$;
   while ($j <= n$){
   $\quad x = x + 1$;
   $\quad j = j \times 2$;
   }

(d) **Algorithm** *WhileLoop2*(n):

$x = 0$;
$j = n$;
while ($j >= 1$){
    $x = x + 1$;
    $j = 2j/3$;
}

(e) **Algorithm** *WhileLoop3*(n):

$x = 0$;
$j = 2$;
while ($j <= n$){
    $x = x + 1$;
    $j = j^3$;
}

(f) **Algorithm** *WhileLoop3*(n):

$x = 0$;
$j = n$;
while ($j \geq 2$){
    $x = x + 1$;
    $j = \sqrt{j}$;
}

(g) **Algorithm** *WhileLoop4*(n):

$x = 0$
$j = n$
while ($j \geq 1$)
    for $i = j$ to n do
        $x = x + 1$
    $j = j - 1$
return x

3. **(15 marks)** Consider the following algorithm:

Algorithm *FactorSum*(A, n):

```
Input: Array A of n real numbers.
S ← 0
for i ← 0 to n − 1
        for j ← i to n − 1
                P ← 1
                for k ← i to j
                        P ← P × A[k]
                S ← S + P
return S
```

(a) Analyze the running time of this algorithm asymptotically.
(b) Improve the running time of the algorithm in a way that reduce its asymptotic time complexity, and analyze the running time of your improved algorithm asymptotically.

4. **(20 marks + 10 bonus marks)** Given an array A of n elements that are integers from 0 to n-1, with any of these numbers appearing any number of times.

   (a) (10 marks) Design an algorithm that finds the repeating numbers in A. For example, let n be 7 and array A be 1, 2, 3, 1, 3, 6, 6, the answer should be 1, 3 and 6. Give the pseudo code description of your algorithm and analyze its time complexity.

   (b) (10 marks) Implement your algorithm using Java. Your program takes any sequence of numbers as the input, outputs the repeating number(s).

   (c) (bonus: 10 marks) Design an efficient algorithm for this task with time complexity $O(n)$ and space complexity $O(1)$.