CS2383 Fall 2021 Assignment 2 (110 marks)
Due Tuesday., Oct. 12, by 5pm.

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

---

1. **(15 marks)** Prove or disprove each of the following statements:

   (a) $n^3 + 8n^2 + 6n + 2$ is $\Theta(n^3)$.

   (b) $8n^2 - 6n + 2$ is $\Theta(n^2)$.

   (c) $2n^2 - 3n + 50$ is $\Theta(n^2)$.

2. **(total 30 marks, 5 marks per question)** Analyze the running time of the following algorithms asymptotically.

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

   $p \leftarrow 1$
   for $i \leftarrow 1$ to n^2 do
       $p \leftarrow p \times i$
   return p

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

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

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

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

(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** *WhileLoop4*(n):

```
x ← 0
j ← n
while (j ≥ 1)
        for i ← 1 to j do
                x ← x + 1
        j ← j − 2
return x
```

3. **(total 15 marks, 5 marks per question)** What does each of the following recursive algorithms do? Analyze their running time asymptotically using recursion trees.

(a) **Algorithm** fun1(n, m)

```
if (n = 0)
        return m;
else
        return fun1(n - 1, n + m);
```

(b) **Algorithm** fun2(n)

```
if (n = 1)
```

return 0;
else
return 1 + fun2($\frac{n}{2}$);

(c) **Algorithm** fun3(A, l, h)
Input: A is an array, l and h are two integers.
if ($l \geq h$)
return;
$minindex \leftarrow l$
$minivalue \leftarrow A[l]$
for ($i \leftarrow l + 1$; $i \leq h$; i++)
if ($minivalue > A[i]$)
$minivalue \leftarrow A[i]$;
$minindex \leftarrow i$;
swap($A[l]$, $A[minindex]$);
fun3(A, $l + 1$, h);

4. **(10 marks)** Given a stack that includes n numbers, write a recursive algorithm to sort the elements in the stack. For example, if the contents of the input stack is: 3 (top), 5, 2, 1, 4, the sorted stack should be 1 (top), 2, 3, 4, 5. Assume that the size of the stack is n, what is the time complexity of your algorithm.

5. **(20 marks)** Write a recursive algorithm that reverses a given integer. For example, if the given number is 12345, the output of your algorithm should be 54321. Analyze its time complexity using a recursion tree. Then describe an algorithm for determining a given number w is palindrome or not. A number is called palindrome if it is equal to its reverse. For example, 1221 is palindrome. Implement your algorithm in Java and hand in the source code via D2L.

6. **(20 marks)** Write a recursive Insertion Sort algorithm that takes an array A of n numbers as input. Analyze its time complexity using a recursion tree. Implement your algorithm in Java and hand in the source code via D2L.