CSE 310 Recitation 1

Objectives:

1. Review on LinkedList data structure and Binary search, linear search, Bubble sort algorithms.

Rules:

1. Except for diagrams, charts or tables, answers MUST be provided in typed form.
2. For grading purposes, do NOT just submit the answers, instead copy each question, and put your answer under it. Unreadable and unclear answers will be graded with 0 points.
3. Submit your recitation on Canvas as a single PDF file.
4. For each recitation, you have 2 attempts to submit, but we will ONLY grade your last submission! It's your own responsibility to make sure that you submit the correct file! We will not accept any submissions through email.
5. **Equipment defects and technological difficulties cannot become excuses for late submission. No late submissions will be accepted!**

Question

1. [3 pts] Suppose that you're given the head pointer H to a single-linked list, and a pointer X to one of the nodes in the list. You're now given a pointer to a new node Y. Write a C/C++ code to insert Y into the given linked list, just after the node X. Assume any reasonable structure for the nodes in the linked list (e.g., each node must have a "next" pointer pointing to the next item on the list).

```
Y->next = X->next;
X->next = Y;
```

2. [3 pts] What function is computed by the function $Mystery(n)$ below? Express your answer as a summation and then give its closed form.

```
Mystery ( n ) {
      Sum = 0;
      for ( i =1; i <= n ; i++)
         for ( j =1; j <= i ; j++)
            sum++;
      return  sum;
}
```

n(n +1)/2

3. [2 pts] For the following Bubblesort algorithm, how many times would the inner for loop iterate (give the closed form)? What is the worst-case running time of Bubblesort?

```
BUBBLESORT(A)
    for i = 1 to A.length-1
        for j = A.length downto i+1          //downto means decrease by 1
            if A[j] < A[j-1]
                exchange A[j] with A[j-1]
```

Total loops = A.length * (A.length + 1) / 2 = ((A.length)^2+A.length)/2

Worst-case = O(n^2)

4. [2 pts] Suppose you have a sorted array $A[1..n]$. What are the number of comparisons to search for the existence of an element x in the given array using binary search?

A = array
1 <= # of Comparisions <= base 2 log(A.length)