CSE 373 Section Handout #6: Midterm Review

1. **Big-Oh and Run Time Analysis**:
   a. Describe the worst case running time of the following pseudocode functions in Big-Oh notation in terms of the variable n. Your answer should be as “tight” and “simple” as possible. *Showing your work is not required*

```
I. void happy (int n, int sum) {
     int k = 1;
     while (k < n) {
         for (int i = 0; i < k; i++) {
             sum++;
         }
         k++;
     }
     for (int j = n; j > 0; j--) {
         sum++;
     }
}

II. void sunny (int n, int sum) {
     for (int i = 1; i < n * n; i++) {
         if (sum > 10) {
             for (int j = 0; j < n; j++) {
                 sum++;
             }
         } else {
             sum++;
         }
     }
}

III. void funny (int n, int sum) {
     for (int k = n; k > 0; k = k/2) {
         for (int j = n; j > 0; j--)
             sum++;
         for (int i = 0; i < n; i++)
             sum++;
     }
}
```

   b. Demonstrate that: 25n +300 is O(n) by finding positive integers c and n0 such that the definition of Big Oh is satisfied. You do not need to prove that these constants satisfy the definition, just specify the constants.

2. **Min Heaps**:

a. As discussed on homework 2, a **three-heap** with n elements can be stored in an array A, where A[1] contains the root of the tree. Draw the **three-heap** that results from inserting **5, 2, 8, 3, 6, 4, 9, 7, 1** in that order into an initially empty **three-heap**

c. Draw the result of doing 1 deletemin on the heap you created in part a

c. What is the big-O worst case running time for a findmin operation in binary min heap containing N elements?

3. **Min Heaps - Build Heap**
    a. Using the **insert** method described in class write pseudocode for a **client implementation** of buildHeap:

    b. Give a tight bound for the worst case asymptotic runtime of your function, in terms of n.

    c. Is this implementation as good as Floyd’s buildHeap? Explain. What is the runtime of Floyd’s buildHeap?

4. Draw the AVL tree that results from inserting the keys **1, 4, 6, 8**, **9, 5,** in that order into an initially empty AVL tree.

5. **Hash-Tables**: For each of the following versions of hash tables, insert the following elements in this order: **55, 86, 16, 25, 6, 7**. For each table, **TableSize = 10**, and you should use the primary hash function **h(k) = k%10**. For double-hashing, using **g(k) = k%7**. If an item cannot be inserted into the table, please indicate this and continue inserting the remaining values.

a. Linear Probing:

0	
1	
2	
3	
4	
5	
6	
7	
8	
9	

b. Quadratic Probing:

0	
1	
2	
3	
4	
5	
6	
7	
8	
9	

c. Separate Chaining:

0	
1	
2	
3	
4	
5	
6	
7	
8	
9	

d. Double hashing:

0	
1	
2	
3	
4	
5	
6	
7	
8	
9	

e.What is the load factor in Table a?

f. What is the load factor in Table b?

g. Describe what happens as the load factor increases for Table a

h. Describe what happens as the load factor increases for Table b.

6. **Design Decisions**

   For each design decision, choose the most efficient structure in terms of time complexity. You may choose from the following:

   Array, sorted array, sorted linked list, AVL Tree, min heap, queue implemented with an array, stack implemented with a linked list.

   Explain your decision. If you use a sorted structure, explain what it is sorted by.

   a)You are in charge of managing an airport terminal's gates, with ID's 1 through 50. Important operations include: marking a gate as in use or unused.

   b) Your task is to store a directory of employees who work at a supermarket. Important operations include the ability to add an employee to the directory, to determine whether someone works at the store (based on name), and the ability to print all of the employees in sorted order. You may assume that no two employees have the same name.

   c) You want to keep track of the 100 most expensive houses in a neighborhood. The only operations that must run fast are adding a new price, and the retrieval of the n'th highest price. For instance, if the following prices are stored: 20, 30, 50, 10, 90, 100 and I ask for the 2nd highest price, 90 would be returned.