CS210 Data Structures (181) **Name____________________________________** **ID__________________**
Major Exam 2 **Weight 15%** **Time: 50 minutes**

There are five questions in this exam. Answer all questions. Time is of the essence, use it wisely!

{CLO 1}Q1 [3]. Consider the following multiple choices, circle the correct answer. Each choice is worth 0.5 points.

A. Consider the below single-left rotation pseudo code for an AVL tree where an AVL node contains value pointers to left, right child nodes.

```
AVLNode single_left_rotation(AVLNode z){
AVLNode x,y,a,b,c,d;
y=z.left;
x=y.left;
a=x.left;
b=x.right;
c=y.right;
d=z.right;
return y;  }
```

What is missing?
a) y.right = z;
b) z.left = c
c) y.right = z; z.left = c;
d) None of the above

B. For the code snippet given in part A, what is the height at node y?
a) Math.max(y.height – x.height) + 1
b) y.height = x.height +1;
c) y.height = z.height +1;
d) None of the above

C. In a binary max-heap implementation using an array, what is the position of a parent node for an arbitrary node x; assume the index starts at 1.
a) (i/2) position
b) (i+1)/ position
c) Math.floor(i/2) position
d) Math.ceil(i/2) position

D. Given an array of element 5,7,9,1,3,10,8,4. Tick all the correct sequences of elements after inserting all the elements in a min-heap.
a) 1,3,4,7,8,9,10
b) 1,4,3,8,9,5,7,10
c) 1,3,4,5,8,7,9,10
d) None of the mentioned

E. What are the worst case and average case complexities of a binary search tree?
a) O(n), O(n)
b) O(logn), O(logn)
c) O(logn), O(n)
d) O(n), O(logn)

F. What is the worst-case time complexity to delete an element from a hash-table implemented with separate-chaining?
a) O(n)
b) O(logn)
c) O(nlogn)
d) O(1)

{CLO 2}Q2. [3]. Sort the array of integers = **{7,5,6,11,10,9,16,15,13,12,8,3}** using quick sort. Show all sequences of <u>**Pivots**</u> during partitioning.

{CLO 3}Q3. [3]. Assume you are using a programming language that allows different data types to be stored in a single array. Given an array of N elements of three different types: Cold, Warm, and Hot; design and describe clearly an **in-place** algorithm to put all the cold elements, on the left, followed by all the warm elements, followed by all the hot elements on the right. Your algorithm can use only a small constant amount of extra space. What is the run time? Show how your algorithm would operate on this array:

0	1	2	3	4	5	6	7	8	9	10
C	**W**	**H**	**C**	**W**	**W**	**C**	**H**	**C**	**W**	**C**

{CLO 4}Q4. [3] Consider a hash table of size 9 that stores entries with integer keys. This hash table uses a double hash function

h(k) = k mod 11

f(h(k)) = k mod 7

Insert, in the given order, entries with keys **8, 11, 18, 22, 28, 13, 25** into the hash table using linear probing to resolve collisions. Show all the work and fill the array **A** accordingly.

A

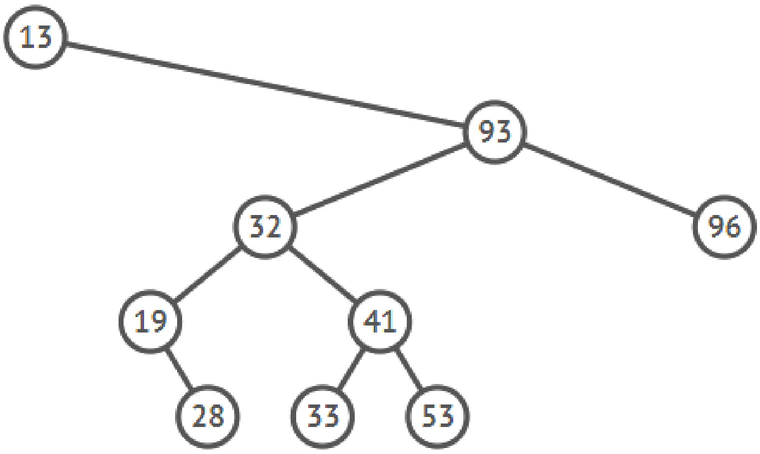
0	1	2	3	4	5	6	7	8

k	*h(k)*	*f(h(k))*	Collisions

What is the number of collisions?

{CLO 1}Q5. [3] Remove the following keys from the Binary search tree given below. Illustrate the tree after each removal. Identify which case applies?

13, 32, 19.



--End of Exam--