CS210 (161) with Dr. Basit Qureshi **Final Exam** **Weight 40%**

Name_____ ID_____

Directions:

- There are 9 questions in this exam. To earn a possible full score, you must solve all questions.
- **Time allowed: 180 minutes**
- Closed Book, Closed Notes.
- There are 10 pages in this exam booklet.
- Use of Calculators and / or computing devices / smartphones etc is strictly prohibited.
- Answer the problems on the exam sheets only. No additional attachments would be accepted.
- Make sure you write your NAME on the paper.
- If you need extra space use the back of a page.
- When the "time is over" is called, it is students responsibility to submit his exam to the invigilator.
- Submitting completed exam 3 minutes after the "time is over" will incur a penalty of 5 points.

Few gentle reminders

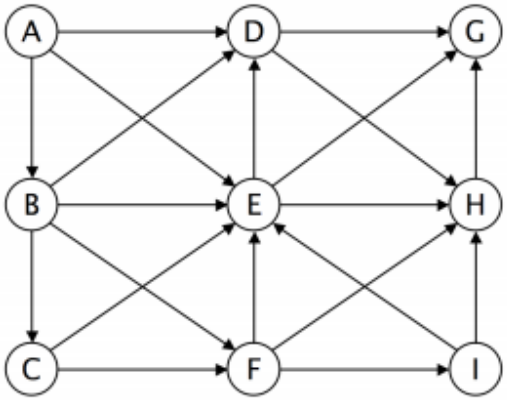
- If you get stuck on some problem for a long time, move on to the next one.
- The ordering of the problems is somewhat related to their relative difficulty. However, the order might be different for you!
- You should be better off by first reading all questions and answering them in the order of what you think is the easiest to the hardest problem.
- Keep the points distribution in mind when deciding how much time to spend on each problem.

Do not write below this line

1 (5 pts)	2 (5 pts)	3 (4 pts)	4 (3 pts)	5 (3 pts)	6 (7 pts)	7 (3 pts)	8 (4 pts)	9 (6 pts)	Total (40 pts)

PART A: CLO 1.

Q1 [5]. Consider the following directed graph for this question



a) Show the adjacency List for the above Directed Graph with vertices sorted in lexicographic order, e.g., when exploring vertex E, consider E-D before E-G or E-H.

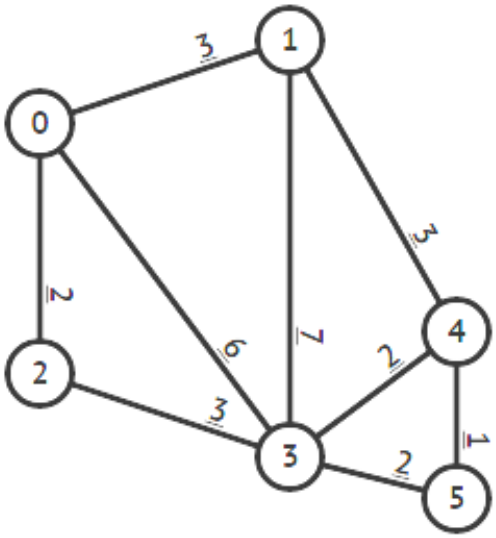
b) Run depth-first search, starting at vertex A. Assume the adjacency lists are in lexicographic order Complete the list of vertices in preorder (the order they are first discovered by DFS). Show the contents of the stack used for DFS traversal.

A – B –

c) Run breadth-first search, starting at vertex A. Assume the adjacency lists are in lexicographic order. Complete the list of vertices in the order in which they are en-queued. Show the contents of the queue used for DFS traversal.

A – B –

Q2. [5] Using Dijkastra's algorithm, compute the shortest path from **1** to all vertices given the following undirected weighted graph. Show the contents of the priority queue and the table of vertices.



PART B: CLO 3

Q3. [4] Consider the following code fragment for a priority queue implemented using a binary heap. Assume `a[]` is an array of integers and $N \geq k \geq 1$.

```
PriorityQueue pq = new PriorityQueue();
int N = a.length;
for (int i = 0; i < N; i++) {
  pq.insert(a[i]);
  if (pq.size() > k) pq.delete();          /* MARK */
}
for (int i = 0; i < k; i++)
  System.out.println(pq.delete());
```

a) What is printed for this code?

b) What is the worst-case scenario, give its running time.

c) Suppose we delete the line commented with /* MARK */, what does it print?

d) Write the output of part (c) using this array.

3	8	2	6	5	1	0	4	7	9

Q4. [3] Consider the following binary tree represented using arrays.

0	1	2	3	4	5	6	7	8	9	10	11	12
Z	W	X	S	T	G	K	F	A	D	B		

a) Draw the binary tree.

b) Show the result of a post-order traversal of this tree

c) Which keys have to be moved so the resulting tree is a Binary Search Tree (BST). Circle the keys.

0	1	2	3	4	5	6	7	8	9	10	11	12
Z	W	X	S	T	G	K	F	A	D	B		

Q5. [3] Draw a complete binary tree given the following information.

- Height of tree must not exceed 4
- Each node in the tree stores only one character
- The pre-order traversal of this tree gives **A L M O N D F U D G E C A K E**
- The in-order traversal of this tree gives **O M N L F D U A E G C D A K E**

PART 3: CLO 2

Q6. [7] Consider the following array of Strings for this question

a) Sort the array using quick-sort algorithm with first value selected as pivot. Assume the values in the array are already shuffled. Identify pivots and show values in the array at each pass.

hash							
fifo							
lifo							
next							
exch							
push							
sort							
swap							
type							
tree							
null							
link							
root							
find							
left							
leaf							

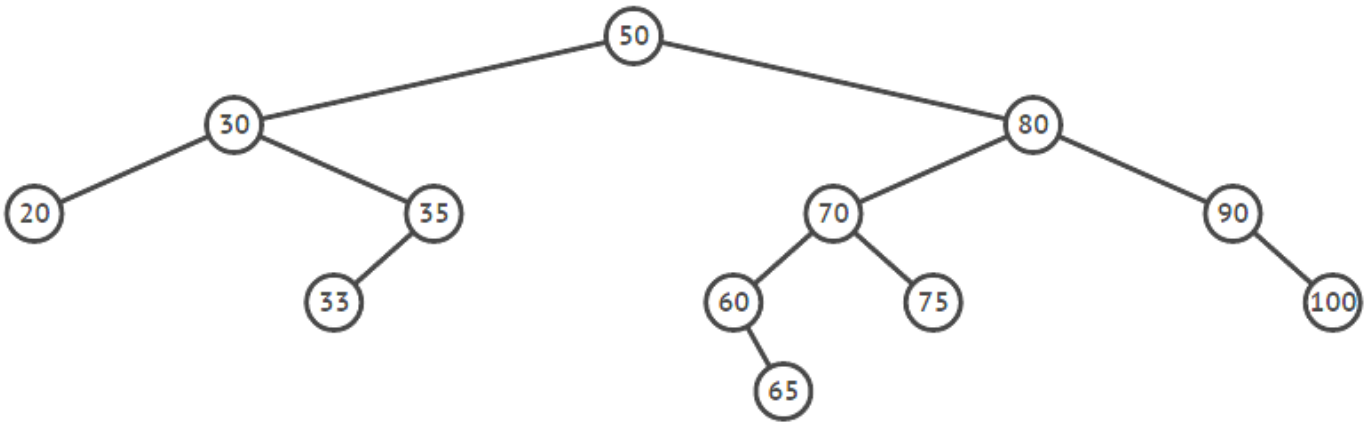
b) Estimate, how many compareTo() and swap() calls would be made using quick-sort?

c) In the worst-case scenario, how many compareTo() and swap() calls would be made using selection-sort?

Q7. [3] You are managing a company called MyBigBankCo that manages an array of Accounts for their customers. Each account contains only two items, String Customer_Name and double balance. You would like to re-arrange the array so that the richest customers (balances greater than 1 million SR) are grouped at the beginning with everyone else at the end.

Describe the algorithm that runs in Linear time using only constant extra space/memory.

Q8 [4]. Observe the following AVL Tree.



a) Insert the following keys and re-draw the tree. Identify appropriate rotations at each step.

68, 34

b) Remove **70** from the tree resulting from part (a) of this question and re-draw the tree.

PART 4: CLO 4

Q9 [6]. A programmer writes a bad hash function to store single-character-keys in a hash table. The bad hash function allows the keys of alphabets (A through M) hash to value 6 in the table and keys of alphabets (N through Z) hash to value 9 in the table.

a) Use linear probing to add the following keys in an initially empty table of size 11

A B A D H A S H I N G

0	1	2	3	4	5	6	7	8	9	10

b) If the size of this array is increased to 2N what will be the average time for searching a random key in this array?

c) Show the hash-table using separate-chaining for the above data

0
1
2
3
4
5
6
7
8
9
10

d) If the size of the array is decreased to N/5 what will be the average time for searching a random key in this array?