CS210 Data Structures (181) Final Exam

Name:____ ID_______________________

Instructions:

- This exam contains four questions with multiple parts.
- Time allowed: 180 minutes
- Closed Book, Closed Notes.
- There are 12 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.
- 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:

Q1	Q2	Q3	Q4	Total:/40

CLO Assessment

CLO1 – Q1(10)	CLO2 – Q2 (10)	CLO3 – Q3 (16)	CLO4 – Q4 (4)

Question1 [CLO 1]: [/10 Points]

1-A). State the runtime requirements in big-O for each of the following code fragments. [/2.0]

	Code Fragment	Running Time in big-O

a	`int a = 0;`
	`for (i = 0; i < n; i++) {`
	`    for (j = 0; j < i; j++) {`
	`           a = a + i + j;`
	`    }`
	`}`	

b	`function(int n) {`
	`    if (n==1)`
	`       return 0;`
	`    else`
	`      return 1 + function(n-1);`
	`}`	

c	`int a = 0, i = n;`
	`do {`
	`    a += i;`
	`    i /= 2;`
	`} while (i > 0);`	

d	`int n = N;`
	`for(int p=0;p<N;p++)`
	`{`
	`  while (1){`
	`    if(n>0){`
	`        n = n/2;`
	`    }else`
	`       break;`
	`  }`
	`  n = N;`
	`}`	

1-B). Read the statement and circle the appropriate choice of data structure. [/2.0]

Problem	Answer
A method that reads a list of names and prints all the names in the opposite order a. Priority Queue b. Stack c. Binary Tree d. minHeap	
A method that arranges the patients names according to the severity of injuries in a hospital emergency room. a. MaxHeap b. Hash table c. Priority Queue d. Queue	
A graph storage mechanism that ensures a constant run-time to find an edge that may or may not exist in the graph. a. Adjacency list b. Adjacency matrix c. Edge list d. None of the above	
A mechanism that allows sorting all courses in your schedule based on the pre-requisite. a. DiGraph b. DAG c. Depth First Search d. Topological sort	

1-C). Draw a binary tree with these conditions:

[/2.0]

Draw a binary tree with these conditions:

Each node in the tree stores only one character
The in-order traversal of this tree gives:
C O T T O N C A N D Y

The post-order traversal of this tree gives:
C O T O N A Y D N C T

1-D). Insert the following keys in a priority queue (minHeap); show all insertions, sink and swim operations.

[/2.0]

10 7 8 25 5 11 2

1-E). Remove two keys from the above heap. Show all sink swim operations

[/2.0]

Question2 [CLO 2]: [__/10 Points]

2-A). Show the sort operations of a bottom-up merge sort for the following keys. Use standard order of alphabets for sorting (A to Z).

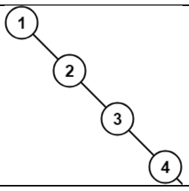
[/4.0]

cat	cam	and	zed	bad	arm	bbd	bam	abd	dad	jar	pat	car

ABCDEFGHIJKLMNOPQRSTUVWXYZ

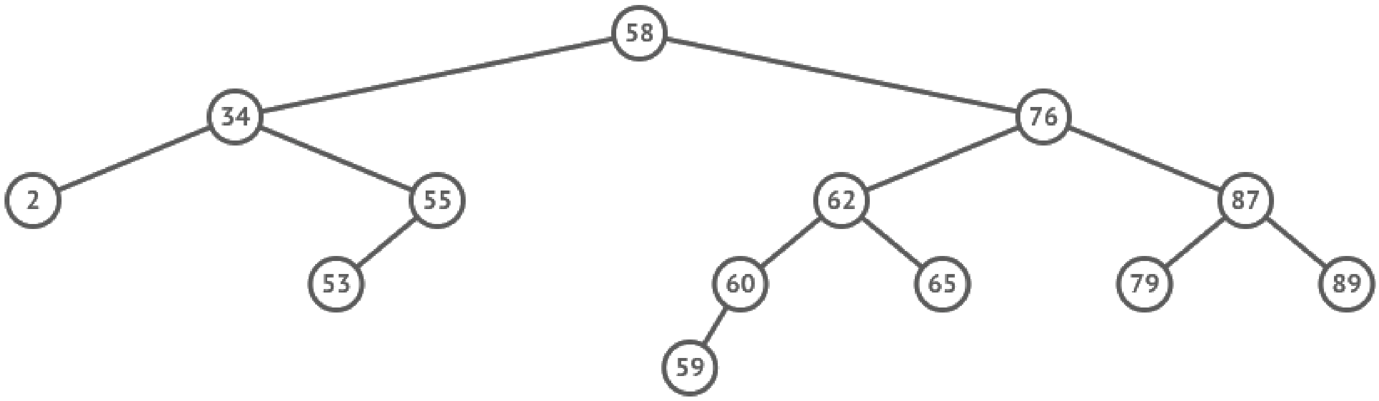
2-B). Read the statement and write the appropriate choice in the box provided

[/3.0]

Problem	Answer
Which sorting algorithm will take the least time (number of comparisons) when all elements of input array are identical? Consider typical implementations of sorting algorithms. a. Insertion Sort b. Heap Sort c. Merge Sort d. Selection Sort	
Assume an array if integers = {2, 4, 3, 1, 6}; applying quick sort, which values would be selected as pivots a. 2 and 1 b. 2 and 4 c. 3 and 6 d. 3 and 1	
Which of the following is not a stable sorting algorithm in its typical implementation. a. Merge sort b. Quick sort c. Insertion sort d. None of the above	
Which sorting algorithm gives $O(n^2)$ comparisons and O(n) swaps for random data? a. Quick sort b. Insertion sort c. Merge sort d. Selection sort	
The run-time for searching a key in a skewed Binary Search Tree is a. O(n) b. O(log n) c. O(1) d. $O(n^2)$  	
Which of the following is an inappropriate search sequence for 36 in a BST composed of integers? a. 93, 27, 34, 62, 99, 36 b. 11, 22, 33, 40, 35, 36 c. 23, 48, 42, 40, 33, 36 d. 11, 12, 13, 14, 25, 36	

2-C). Identify all rotations to remove__2__ from the following AVL Tree.

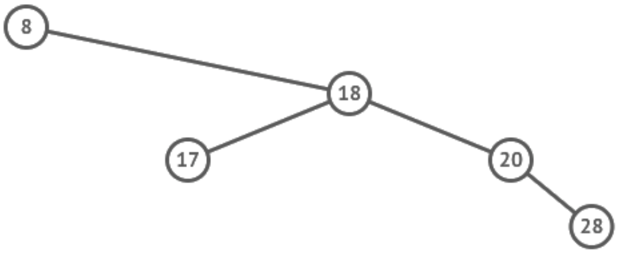
[/3.0]



<u>Question3 [CLO 3]: [___/16 Points]</u>

3-A). Give pseudocode for a method called <u>ancestors</u> for a BST node P. This BST stores nodes with keys of type integers. It prints all the parent, grand-parent, grand-grand-parent etc keys on console. If a P has no ancestors, it prints -1.

[/3.0]



Example: Ancestors of P=28 are: 20, 18, and 8.

3-B). Give code for a method called degree. This method takes two parameters, an undirected Graph G represented by a 2-dimensional Array (Adjacency matrix) and an integer V representing a vertex in graph G. This method counts and returns the number of edges connected to vertex V.

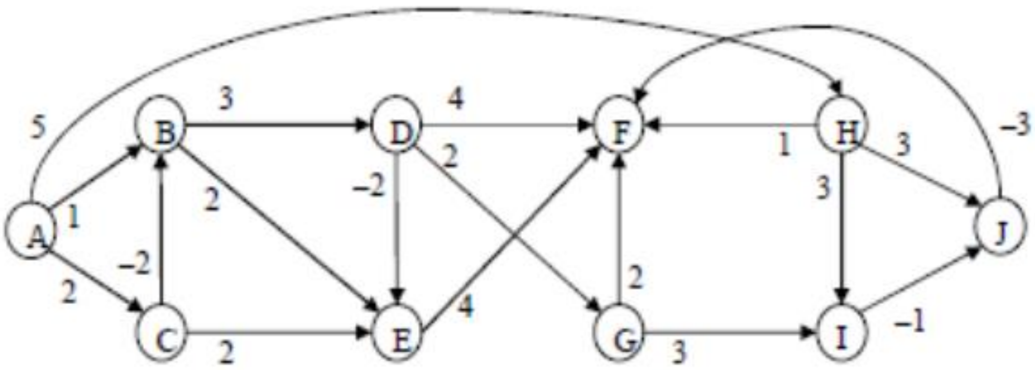
[/3.0]

```
public int degree(Graph G, int v) {
```

	1	2	3
1	0	1	0
2	1	0	1
3	0	1	0

3-C). For the following DiGraph provide an Adjacency list.

[/2.0]



A
B
C
D
E
F
G
H
I
J

3-D). For the above graph, show a Breadth First Search (BFS) run starting at A. Consider lowest weights for traversal. Use appropriate Data Structures.

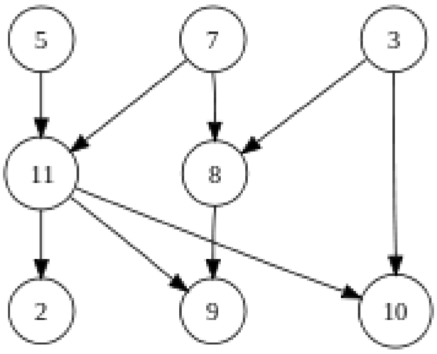
[/2.0]

3-E). Given a DiGraph G with V vertices and E edges. Give a brute force algorithm that finds all possible paths from a source S to a destination D using BFS. What is the run-time for this algorithm?

[/3.0]

3-F). For the graph shown below, provide the result of a DFS based topological sort algorithm starting at 3. Show the content of the extra stack used.

[/3.0]



<u>Question4 [CLO 4]: [__/4 Points]</u>

4). Given input {4373, 1323, 6173, 4194, 4345, 9673, 1983, 2106, 3909}, a hash table A of size 10 and a hash function h(x) = x mod 10, show the resulting hash map using each of the following collision resolution methods:

[/4.0]

Separate chaining hash table.

A

x	h(x) = x mod 10

0	
1	
2	
3	
4	
5	
6	
7	
8	
9	

Hash table using linear probing.

	0	1	2	3	4	5	6	7	8	9
A										

How many collisions?

How many displacements?

--End of Exam--