CS210 Data Structures (191) 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 < n; j++) {`
	`  a = a + i + j;`
	`  }`
	`}`	

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

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 data structure that can be used to read a list of names and prints all the names in the opposite order a. Priority Queue b. Binary Tree c. Stack d. minHeap	
A method that arranges the patients names according to the severity of injuries in a hospital emergency room. a. Stack b. Priority Queue c. Hash table d. Queue	
A graph storage mechanism that ensures a constant run-time to find an edge (source-destination) that may or may not exist in the graph. a. Adjacency matrix b. Adjacency list c. Edge list d. None of the above	
Which data-structure is most appropriate for searching in terms of time complexity? a. Hashtable b. Doubly Linked List c. Tree d. Array	

1-C). Draw <u>a</u> 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 (LVR) of this tree gives:
R A S P B E R R Y P I

The post-order traversal (LRV) of this tree gives:
R A S B E R I P Y R P

The height of the tree does not exceed 5 including the root (counting the number of nodes in the longest path).

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

[/2.0]

10 7 8 25 5 11 26

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

[/2.0]

Question2 [CLO 2]: [/10 Points]

2-A). Show the sort operations of a quick sort algorithm for the following keys. Identify all keys that would be swapped in addition to the pivots in all iterations.

[/4.0]

3	6	5	4	1	2	10	8	7	15	13	12	11

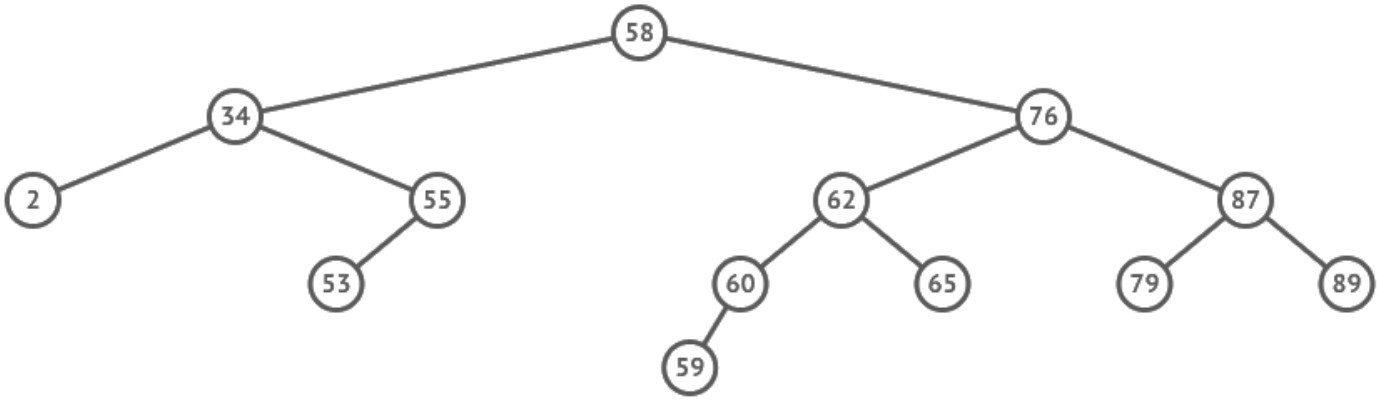
2-B). Read the statement and write the appropriate choice in the box provided

[/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. Selection Sort c. Heap Sort d. Merge 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 2	
Which of the following sorting algorithms requires additional memory of size *n*. 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 an array with random data? a. Quick sort b. Insertion sort c. Merge sort d. Selection sort	
The run-time for searching a key in a complete 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 34 from the following AVL Tree.

[/3.0]

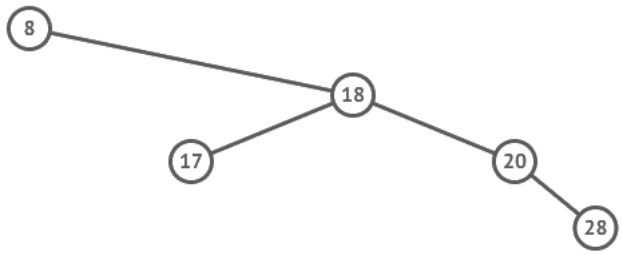


Question3 [CLO 3]: [/16 Points]

3-A). Write a recursive method ancestors that takes a BST node as a parameter. The method prints all the ancestors (parent, grand-parent, grand-grand-parent etc) keys on console. If a node has no ancestors, it prints -1. The BST Node class is given below for your references

[/3.0]

```
public class Node{
  int V;
  Node left;
  Node right;
  Node parent;
  public Node(){
    V = 0;
    left = right = parent = null;
  }
}
```



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

```
public void ancestors (Node P){
```

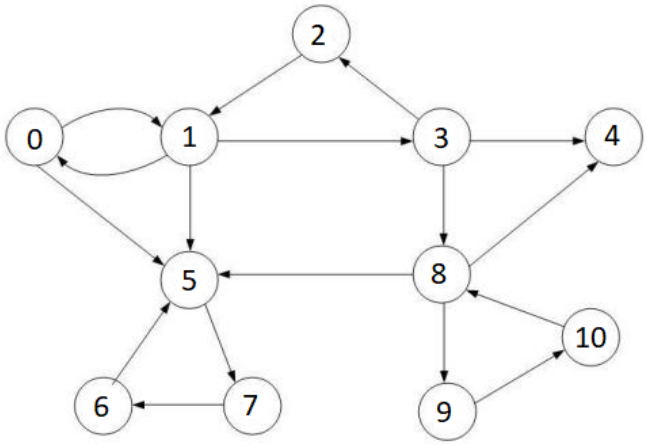
3-B). Write java code for a method called <u>outdegree</u> that takes two parameters, an undirected Graph G represented by an <u>Adjacency-list</u> and an integer V representing a vertex in graph G. This method counts and returns the number of out-directed edges connected to vertex V.

[/3.0]

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

3-C). For the following Directed Graph, provide an Adjacency list.

[/2.0]



0
1
2
3
4
5
6
7
8
9
10

3-D). For the above graph, show a Depth First Search (DFS) run starting at <u>0</u>. Consider the order of nodes (0 comes before 1) when choosing the next edge in the path. Use appropriate Data Structures to show your work.

[/2.0]

3-E) Assume that you are given a singly linked list with sorted data (Integer values) in ascending order; Write a method `void insertSorted(int X)` that takes integer X as a parameter and inserts a new node containing this value at the appropriate location in the list. After the insertion, the sorted order should not change and the list remains sorted. Give a Big-Oh notation for the worst-case runtime.

[/4.0]

```
void insertSorted(int X){
```

```
public class Node {
  int value;
  Node next;
  public Node(){
    value = 0;
    next = null;
  }
  public Node(int v, Node nxt){
    value = v;
    next = nxt;
  }
}
```

3-F) Write short answers for the following

[/2.0]

Explain why the time complexity of inserting an item into a unsorted linked list is sometimes, but not always, better than the time complexity of inserting an item into a hash table
Explain why searching for an item in a binary heap gives O(n) time whereas an AVL tree gives O(log n) time.

Question4 [CLO 4]: [/4 Points]

4-A). A linear probing hash table A of size 10 and a hash function h(x) = x mod 10 is used to insert six integer keys into an initially empty hash table; the array of the keys is shown below:

[/4.0]

Which if the following choice(s) are insertion sequences resulting in this table. Assume that the size of the hash table does not change during the insertions.

0	
1	
2	**42**
3	**23**
4	**34**
5	**52**
6	**46**
7	**33**
8	
9	

A. 46 42 34 52 23 33
B. 34 42 23 52 33 46
C. 46 34 42 23 52 33
D. 42 46 33 23 34 52
E. 42 23 34 52 46 33

4-B). How many collisions (probes) when inserting 52?

4-C). A separate chaining strategy is used to avoid collisions in a hash table B. This has table is of size 10 using a hash function h(x) = x mod 10; show how would you insert the following keys in this hash table.

1011, 1010, 1001, 1000, 1100, 1101, 1110, 1111

0	
1	
2	
3	
4	
5	
6	
7	
8	
9	

4-D) What is the problem with the above hash function, propose a solution for this issue.

--End of Exam--