National University of Computer & Emerging Sciences, Karachi
Final Examination (Fall-2023)
Dec 18, 2023, 12:30 PM – 3:30 PM

Course Code: CS2001	**Course Name: Data Structures**
Instructor Name: Dr. Jawwad A. Shamsi, Dr. Fahad Sherwani, Mr. Basit Ali, Ms. Sobia Iftikhar, Ms. Mubashra Fayyaz, Mr. Minhal Raza, Dr. Anam Qureshi, Ms. Sumaiyah Zahid, Ms. Sania Urooj, Mr. Farooq Zaidi, Mr. Shahbaz Siddiqui, Ms. Fizza Mansoor, Dr. Farrukh Hasan	
Student Roll No:	**Section No:**

Instructions:

- Return the question paper and make sure to keep it inside your answer sheet.
- Read each question completely before answering it. There are **6 questions in 4 pages**.
- In case of any ambiguity, you may make assumption. But your assumption should not contradict any statement in the question paper.
- You are **not allowed to write** anything on the question paper (except your ID and group).

Time: 180 minutes. **Max Marks**: 100 Points

Question: 1 [20 points] [CLO 2]

True/ False with reasoning. (Note: You need to give not more than 3 lines justification for each irrespective of whether the statement is true or false. There will be NO marks without justification) [20 points]

1. Is it possible to find the middle node of a singly linked list in constant time (O(1))?
2. Can a stack be efficiently implemented using two queues, and vice versa?
3. In a sorted array of distinct integers, a linear search is more efficient than a binary search for finding a specific element.
4. Among the bubble sort, insertion sort, and selection sort algorithms, selection sort is the most likely to have the best performance on a nearly sorted array because it is an adaptive algorithm.
5. The standard versions of both Merge sort and quick sort are both in-place sorting algorithms, allowing them to sort data without requiring additional memory space.
6. A balanced binary tree is also a complete binary tree.
7. In an AVL tree, after the insertion of a node, the tree may require rotations to maintain its balance factor and ensure that the height difference between the left and right subtrees is at most 2.
8. A binary heap is a suitable data structure for implementing a priority queue because it ensures that the element with the highest priority is always at the root.
9. In a weighted graph, where the edge weights are non-negative, applying Dijkstra's algorithm and Prim's algorithm for finding the minimum spanning tree will always yield the same result.
10. A hash table with a higher load factor is generally more efficient because it maximizes the utilization of available memory space.

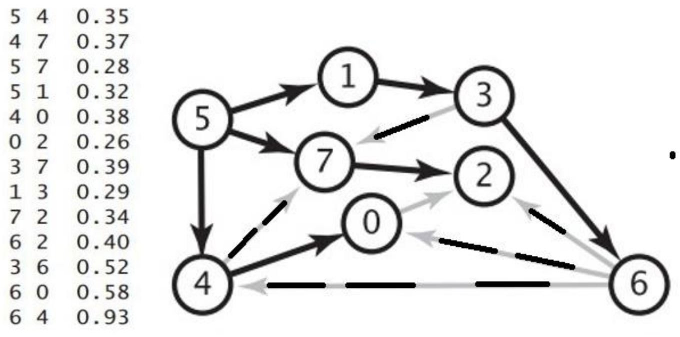
Question: 2 [10 points] [CLO 1]

You are given an array of size ***N*** containing distinct integers. An inversion in an array occurs when there are two indices i and j such that i < j but arr[i] > arr[j].
Your task is to implement an either algorithm or (pseudo-code) better than O(n^2) to count the number of inversions in the array efficiently, where a positive integer N represents the size of the array of ***N*** distinct integers and is entered as input by the user.
Example:
Input:
N = 6

Array = [5, 3, 9, 1, 7, 6]
Output:
(5, 3) is an inversion because 5 > 3.
(5, 1) is an inversion because 5 > 1.
(9, 1) is an inversion because 9 > 1.
(7, 6) is an inversion because 7 > 6.
(9, 7) is an inversion because 9 > 7.
(9, 6) is an inversion because 9 > 6.
(3, 1) is an inversion because 3 > 1.
So, there are a total of 7 inversions in the given array. The problem solved above in O(n^2) time. Find an another way to do it in more efficient manner.

Question: 3 [10 points] [CLO 4]

Consider the following directed graph ***G :<V,E>*** as shown in figure-2. The 1st column shows the source vertex, 2nd column shows the destination vertex and 3rd column shows the cost between source and destination vertex.



5	4	0.35
4	7	0.37
5	7	0.28
5	1	0.32
4	0	0.38
0	2	0.26
3	7	0.39
1	3	0.29
7	2	0.34
6	2	0.40
3	6	0.52
6	0	0.58
6	4	0.93

Figure 1

a. Perform topological sorting on the graph. Show each step clearly by showing the nodes in the underlying data structure at each stage. [5 points]
b. Construct the minimum spanning tree (MST) for the given graph using Kruskal's Algorithm. Show each step clearly. [5 points]

Question: 4 [20 points] [CLO 3]

In Josephus problem, ***N*** people stand in a circle waiting to be executed. The counting starts at some point in the circle and proceeds in a specific direction around the circle. In each step, a certain number of people are skipped and the next person is executed (or eliminated). The elimination of people makes the circle smaller and smaller. At the last step, only one person remains who is declared the 'winner'.

Therefore, if there are ***N*** number of people and a number ***K*** which indicates that ***K–1*** people are skipped and ***k–th*** person in the circle is eliminated, then the problem is to choose a position in the initial circle so that the given person becomes the winner.

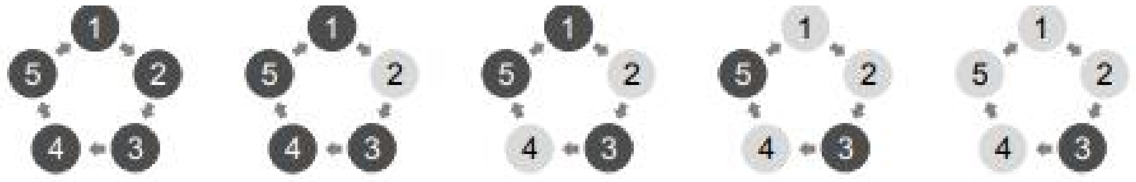


Figure 2

For example, if there are 5 (***N***) people and every second (***K***) person is eliminated, then first the person at position 2 is eliminated followed by the person at position 4 followed by person at position 1 and finally the person at position 5 is eliminated. Therefore, the person at position 3 becomes the winner.

Try the same process with N = 7 and K =3. You will find that person at position 4 is the winner. The elimination goes in the sequence of 3, 6, 2, 7, 5 and 1. Write a program which finds the solution of Josephus problem using a circular linked list.

Question: 5 [20 points] [CLO 4]

The following figure-3 describes a breadth first search (BFS)

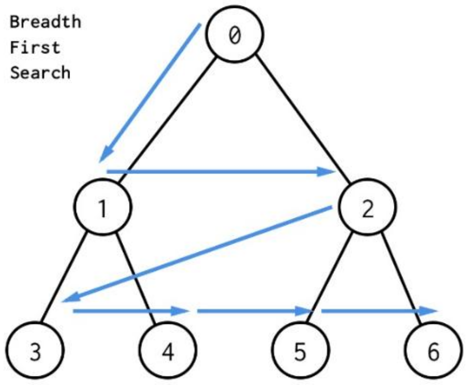


Figure 3

The nodes in an integer binary tree can be represented by the following data structure.

```
class BTNode { // Binary tree node
public:
int item; // data in this node
BTNode left; // left subtree or null
BTNode right; // right subtree or null if none
}
```

Give two implementations (code) of the definition of method BFS below to perform a breadth-first traversal of a binary tree and print the node data values in the order they are reached during the search.

```
// Perform a breadth-first traversal of the binary tree with
// root r and print the node values as they are encountered
public void BFS(BTNode r);
```

1. In your first solution, you may only use recursion and no other data structure in your implementation [10 points]
2. In your second solution you may create and use instances of other standard data structures (array's or vector's) and their operations if they are useful, without having to give details of their implementations. [5 points]
3. What will be the time complexity of both solutions?List pros and cons of both. [5 points]

Question: 6 [20 points] [CLO 3]

ABC	DFD	*LMN*	*PQR*
AAA	BBB	*TRS*	*XZY*
ZZG	LMN	PQR	GGG
UVT	SRT	XYZ	PPP

Table-1

MATCHED

LMN	*PQR*
SRT	*XYZ*

Table-2

You are given a text based grid shown as Table-1 where you need to match a small grid as shown in Table-2 with a specific pattern .

a. Dry run the algorithm and find the maximum number of matching grids. The hash function for an individual cell is defined as H(cell)=(Char1 ASCII value +Char2 ASCII value+Char3 ASCII value)%13 and hash function for a grid will be H(grid)= (H(cell1) +H(cell2)+H(cell3)+H(cell4))%13.

b. Write a code that will use rolling of a hash function to check if pattern is matched or not. H(Grid)=(H(Grid)-H(cell removed) mod11) + H(cell added)mod13.