Ref. No.: Ex/IT/PC/B/T/211/2023

B.E. Information Technology Second Year First Semester 2023

Sub: Data Structures and Algorithms

Time: 3 Hours Full Marks: 100

Answer all Questions

1. Answer either (a) or (b)

a. (3 + 4 + 8 + 10)

i. Let A be a four dimensional array declared as follows:

A: array [1...20][1...15][1...10][1...30] of integers;

Assuming that each integer takes one memory location, array is stored in **Column-major** order, and the first element of the array is stored at location 1000, what is the address of the element A[2][3][4][5]?

ii. Derive the time complexity of the following code in big-Oh notation?

```
int fun (int n)
{
        int count = 0;
        for (int i = n; i > 0; i /= 2)
                for (int j = 0; j < i; j++)
                count += 1;
        return count;
}
```

iii. Write an algorithm to print all the LEADERS in a list of numbers. An element is leader if it is greater than all the elements to its right side. And the rightmost element is always a leader. For example in the array {16, 17, 4, 3, 5, 2}, leaders are 17, 5 and 2.

iv. Propose a linked list data structure which allows insertion at the beginning and deletion of nodes from the middle in O(1) time. Write down the insert and delete functions for your data structure.

b. (3 + 4 + 6 + 4 + 8)

i. Let A be a six dimensional array declared as follows:

A: array [1...10][1...10][1...10][1...20] [1...20] [1...20] of integers;

Assuming that each integer takes one memory location, array is stored in **row-major** order, and the first element of the array is stored at location 1000, what is the address of the element A[3][4][3][4][3][4]?

ii. Derive the time complexity of the following codes in Big-C notation.

```
int x=0;
for (j=1; j<n/2; j++)
    for (k=1; k<n^2; k++)
        x=x+j+k;
```

iii. Suppose that each row of an n x n array A consists of 1's and 0's such that, in any row i, all the 1's come before any 0's in that row. Suppose further that the number of 1's in row i is at least the number in row i + 1, for i=0, 1, …. , n-2. Write an O(n) algorithm to count the number of 1's in A.

iv. Consider a sorted linked list having following nodes:

1→2→3→4→5→7→8

You are given pointer to node 7 and a new node having value 6. Can you insert node 6 in O(1) time in the list maintaining the ascending order.

[Turn over

v. Suppose n petrol pumps are connected in a circular linked list. Each node contains two sets of data: The amount of petrol that the petrol pump has and distance from that petrol pump to the next petrol pump. Write an algorithm to calculate the first point from where a truck will be able to complete the circle (The truck will stop at each petrol pump and it has infinite capacity). Assume for 1 liter petrol, the truck can go 1 unit of distance. For example, let there be 4 petrol pumps with amount of petrol and distance to next petrol pump value pairs as {4, 6}, {6, 5}, {7, 3} and {4, 5}. The first point from where truck can make a circular tour is 2nd petrol pump.

(6 + 4 + 4 + 6)

a. A circularly linked list is used to represent a Queue. A single pointer variable p is used to access the Queue. To which node should p point such that both the operations EnQueue and DeQueue can be performed in constant time? Write down the EnQueue and DeQueue functions.

a. Consider following code to calculate factorial of a given number:

```
int fact (int n)
{
        if (n ==0)
                return 1;
        else
                return (n*fact (n-1));
}
```

Show the content of the stack when fact (4) is called.

b. The following postfix expression with single digit operands is evaluated using a stack:

8 2 3 ^ / 2 3 * + 5 1 * -

Note that ^ is the exponentiation operator. What are the top two elements of the stack after the first * is evaluated?

c. Let S be a stack of size N ≥ 1. Starting with the empty stack, suppose we push the first n natural numbers in sequence, and then perform n pop operations. Assume that Push and Pop operation take X seconds each, and Y seconds elapse between the end of one such stack operation and the start of the next operation. For M ≥ 1, define the stack-life of M as the time elapsed from the end of Push(M) to the start of the pop operation that removes M from S. Show that the average stack-life of an element of this stack is N * (X + Y) - X.

Answer either (a) or (b)

a. (5 + 4 + 7 + 6)

i. Write a recursive function to compute the height of a binary tree.

ii. A priority queue is implemented as a Max-Heap. Initially, it has 5 elements. The BFS traversal of the heap is: 10, 8, 5, 3, 2. Three new elements 1, 7, and 9 are inserted into the heap in that order. Give the BFS traversal of the heap after the insertion of the elements.

iii. Insert the following keys in an AVL tree.

10, 85, 15, 70, 20, 60, 30, 50, 65, 80, 90, 40, 5, 55

iv. Show that for a B tree of order m with height h and n number of keys, the following inequality holds.

$$\log_m(n+1) - 1 \le h \le \log_{\frac{m}{2}}\left(\frac{n+1}{2}\right)$$

b. (9 + 5 + 8)

i. Write an algorithm to convert a binary search tree into a sorted doubly linked list in place. Think of the left child and right child pointers as synonymous to prev and next pointers of doubly linked list.

ii. The in-order traversal sequence of a binary search tree is 2, 3, 4, 5, 6, 8, 9, 10, 12, 15, 20 and the post-order traversal sequence of the same binary search tree is 2, 4, 3, 6, 9, 8, 5, 12, 20, 15, 10. Reconstruct the original binary search tree.

iii. Insert the following keys into an empty B tree of order 5 and finally delete key 7.

1 12 8 2 25 6 14 28 17 7 52 16 48 68 3 26 29 53 55 45

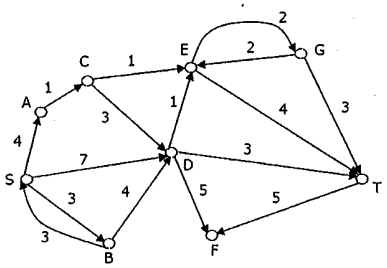
4. (2 + 5 + 8)

a. Match the algorithms in group A with the data structures in group B.

Group A	Group B
DFS	Queue
BFS	Priority Queue
Prim's Algorithm	Heap
Dijkstra's Algorithm	Stack

b. Write an algorithm (other than Kruskal's and Prim's algorithm) to find a spanning tree of a given undirected graph.

c. Consider the directed graph shown in the figure below. There are multiple shortest paths between vertices S and T. Which one will be reported by Dijstra's shortest path algorithm? Assume that, in any iteration, the shortest path to a vertex v is updated only when a strictly shorter path to v is discovered.



5. Answer either (a) or (b)

a. (6 + 6 + 4 + 2)

i. Suppose we are sorting an array of eight integers using quicksort, and we have just finished the first partitioning with the array looking like this: 2 5 1 7 9 12 11 10. Complete the sorting using Quicksort algorithm.

ii. It is required to search an arbitrary element to find the element closer to a given number. Give an algorithm based on the principle of linear search. Assuming that the given number is not equal to any array element, find the expected number of comparisons.

iii. Show that the time complexity for merge sort algorithm is O(nlogn).

iv. Consider a hash table with 100 slots. Collisions are resolved using chaining. Assuming simple uniform hashing, what is the probability that the first 3 slots are unfilled after the first 3 insertions?

b. (5 + 3 + 6 + 4)

i. You are given a linked list of 0's, 1's, and 2's. Write an algorithm to sort the linked list.

ii. Suppose P, Q, R, S, T are sorted arrays having lengths 20, 24, 30, 35, 50 respectively. They are to be merged into a single array by merging together two arrays at a time. What is the number of comparisons that will be needed in the worst case by an optimal algorithm for doing this?

iii. Run the heap sort algorithm on the following array to arrange the numbers in decreasing order.

50 30 60 10 40 20 90 80 100 70

iv. Consider a hash table of size seven, with starting index zero, and a hash function (3x + 4) mod 7. Assuming the hash table is initially empty; insert the sequence 1, 3, 8, 10, 4, 6, 12, 2 into the hash table using open addressing with linear probing.