VIT
Vellore Institute of Technology

Final Assessment Test – November/December 2023

Course: BCSE202L - Data Structures and Algorithms
Class NBR(s): 5396 Slot: E1+TE1
Time: Three Hours Max. Marks: 100

KEEPING MOBILE PHONE/SMART WATCH, EVEN IN 'OFF' POSITION, IS TREATED AS EXAM MALPRACTICE

Answer any TEN Questions

(10 X 10 = 100 Marks)

1. Calculate the time complexity for the following expressions by iteration method:

   i) T(n)=2T(n/2) + n2, T(1)=1 [3]
   ii) T(n)=4T(n/2) + n, T(1)=1 [3]
   iii) T(n)= 2T(n-2) + c, T(0)=1 [4]

2. Generate the relation/expression of time complexity for the following piece of code and subsequently calculate its time complexity.

```
int oper(int a[], int n, int s)
{     if (n <= 0)
        return (-1);
      if (a[n-1] == s)
        return (n-1);
      else
        return oper (a,n-2, s);
}
```

3. A stack ADT operates by push and pop functions with top being the index pointer. A queue ADT operates by enqueue and dequeue operations with front and rear index pointers. Implement a queue by using a stack underneath (with array). This means that the enqueue and dequeue functions in turn should call relevant push and pop operations to implement the queue ADT. Write the pseudocode for this enqueue and dequeue operations and justify their corresponding time complexities.

4. Generate the contents of the original array after 7th call to the modified merge sort function is completed. The input array and the modified merge sort are given below:

Input array : 43 39 31 35 42 28 20 27 30 25

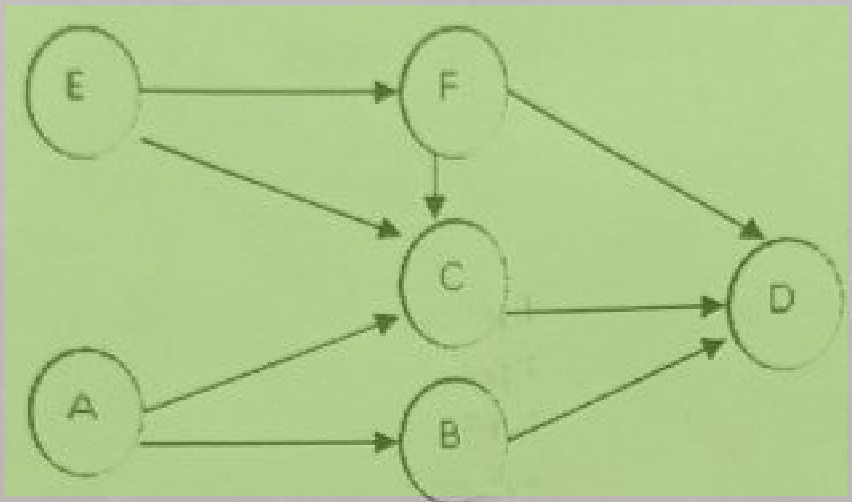
```
void modmergesort (int arr[], int l, int r)
{  int m;
   if (l < r)
   {     m= (l+r) / 2;
        modmergesort(a,m+1,r);
         modmergesort(a,l,m);
         merge(l,m,r);
   }
}
```

5. Apply stack to convert the given infix expression to its postfix expression. '^' represents exponentiation.

A/B^C*((D-E)+F)

Consequently, use the postfix notation to create an expression tree and then perform an inorder traversal of the created expression tree.

6.



Perform topological sorting on the above directed graph.

Now, for performing Depth First Traversal, assume that all the edges in the graph are undirected. Generate, step by step, the resultant output of Depth First Traversal, along with the finishing time for all the vertices, starting from the vertex A.

NOTE: The adjacency matrix/list has the following order A, D, E, F, C, B.

7. The cable company has to put down the cables which can connect all the six houses (A, B, C, D, E, F) and it wishes to minimize the cost of cabling. Use Prim's algorithm (using A as the source) to show in a step by step manner how this can be achieved. The resultant connections and the total cost should also be provided. The cost matrix for the houses is given below. A positive value in (i,j) cell represents a cabling cost between house i and house j, 0 means no cost and inf signifies no cabling can be done.

Houses	A	B	C	D	E	F
A	0	10	30	inf	inf	Inf
B	10	0	25	5	inf	Inf
C	30	25	0	6	25	6
D	inf	5	6	0	inf	13
E	inf	inf	25	inf	0	8
F	inf	inf	6	13	8	0

8. Consider a hashing mechanism which accepts integer keys, k, of maximum six digits, calculates the hash table size, p, as the minimum prime number which is larger than the number of keys entered (for eg., if the number of entries are 5, the hash table size p is 7). Using the right 2-shift method for hashing in which two right most digits are rotated to the left and then use digit extraction (first, third, and fifth digit) get the addresses of the following keys. Use the linear probe method to resolve collision and report total number of collisions occurred during the process.

224562 137456 214562 140145
214576 162145 144467 123456

9. Evaluate the performance (in terms of no. of collisions) of general modular function and subsequent quadratic probing for the following keys having five digits. If the digit has less than five digits, perform padding on the right side (least significant bits). The hash table size, p, is the minimum prime number larger than the length of the input list.
The numbers are 2543, 71356, 25328, 23151, 131, 4217, 99929, 12345

10. Demonstrate how a min heap can be used to satisfy the operations of a priority queue. The priority queue always removes the node with the lowest value (highest priority) while enters a node simply at the end. Simulate schematically how the following entries are added/enqueued (and adjusted) in this priority queue by using a min heap. Show the status of insert enqueue operation.

12, 18, 11, 7, 31, 20, 45, 14 and 32

Subsequently perform remove/dequeue operation twice. Show the adjustments and the subsequent status of the min heap tree after each remove/dequeue operation.

11. Create an AVL tree by inserting the following sequence of values while maintaining the height balance of the binary search tree :

10 20 30 40 8 6 4 2

Consequently, delete the keys 8 and 4 in sequence from the generated tree. Reveal the status of the tree after each insertion and deletion operation.

⇔⇔⇔