MANIPAL UNIVERSITY JAIPUR

Registration Number: _____

SOLUTION

FACULTY OF ENGINEERING

SCHOOL OF COMPUTING AND IT

B. Tech (CSE/IT/CCE)

III Semester End Term Examination - 2019-20

CS1303– Data Structures

(CLOSED BOOK)

Duration: 3 Hours **Max. Marks: 80**

Instructions:

- Attempt any five questions.
- Missing data, if any, may be assumed suitably.
- Function/Programs shall be written in programming language 'C'
- Calculators are not allowed.

1. a. Write a non-recursive function to search an element *E* in a single dimensional array *A* of *N* integers using binary search technique. The function shall return (to the calling function) the index position of the element found, if the element exists in the array, otherwise the function shall return -1. The function prototype for searching is as below:

int binarySearch(int A[ ], int N , int E) [8]

Answer:

```
int binarySearch(int A[],int N, int E)
{
        int lb,ub,mid;
        lb=0;ub=N-1;
        while(lb<ub)
        {       mid=(lb+ub)/2;
                if (E==A[mid])  return mid;
                else    if (E>A[mid])   lb=mid+1;
                else    ub=mid-1;
        }
        return -1; }
```

b. Consider a matrix of integers of size m x n. Write a function to display integers which are perfect square in the given matrix.

In mathematics, a square number or perfect square is an integer that is the square of an integer; in other words, it is the product of some integer with itself. For example, 16 is a perfect square, since it can be written as 4 × 4. [8]

Answer:

```
void perfect(int a[][4], int m, int n)
{
        int i,j,k;
        for(i=0;i<m;i++) {
                for(j=0;j<n;j++) {
                        for(k=1;k<a[i][j]/2;k++){
                                if ((k*k)==a[i][j])             {
                                        printf("%d\t",a[i][j]);  break;  } }} } }
```

2. a. Write a function to reverse a singly linked list using pointers (Do not reverse the linked list by swapping the contents). The function prototype for reversing the linked list is as below:

NODE * reverse(NODE * start);

Where NODE is a type which can hold an integer and can store address of the next node of the linked list. [8]

Answer

```
NODE *reverse(NODE *start)
{
        NODE *prev=NULL,*current,*next;
        current=start;
        while(current!=NULL)
        {
                next=current->next;
                current->next=prev;
                prev=current;
                current=next;
        }
        return prev;
}
```

b. Write a function to evaluate a postfix expression *postExp*. Assume that the expression contains single digits operands. The function prototype for evaluating postfix expression is as below:

int postfix(char postExp[ ]);

Assume that there exists a built-in data structure STACK, push and pop operations are already defined. The prototype of push and pop functions are as below:

void push(STACK *s, int digit);
int pop(STACK *s); [8]

Answer

```
int postfix(char postExp[ ])
{
        int op1,op2,i,r;
        for(i=0;postExp[i]!='\0';i++)
        {
                if(postExp[i]>='0'&&postExp[i]<='9')
                {
                        op1=postExp[i]-48;
                        push(&s,op1);
                }
                else
                {
                        op2=pop(&s);
                        op1=pop(&s);
                        switch(postExp[i])
                        {
                                case '+' :      r=op1+op2; push(&s,r);
                                                break;
                                case '-' :      r=op1-op2;push(&s,r);
                                                beak;
```

```
                    case '*':   push(&s,op1*op2);
                                break;
                    case '/':   push(&s,op1/op2);
                                break;
                }
            }
        }
        r=pop(&s);
        return r; }
```

3. a. Convert the following infix expression into prefix expression using STACK. Show status of STACK at each step.

A/(B+C* D/E*F) + G/H [6]

Answer

Reverse of the string is H/G+)F*E/D*C+B(/A

Symbol	Stack	Prefix Expression
H	-	H
/	/	H
G	/	HG
+	+	HG/
)	+,)	HG/
F	+,)	HG/F
*	+,),*	HG/F
E	+,),*	HG/FE
/	+,),*,/	HG/FE
D	+,),*,/	HG/FED
*	+,),*,/,*	HG/FED
C	+,),*,/,*	HG/FEDC
+	+,),+	HG/FEDC*/*
B	+,),+	HG/FEDC*/*B
(	+	HG/FEDC*/*B+
/	+,/	HG/FEDC*/*B+
A	+,/	HG/FEDC*/*B+A
		HG/FEDC*/*B+A/+
		+/A+B*/*CDEF/GH

b. Write a function to create a queue (implemented using linked list) by using elements popped from the stack (implemented using linked list). The top element of the stack shall be the first element of the queue and the bottom element of the queue shall be the last element of the stack. While creating queue, memory shall not be allocated for nodes again. [10]

Answer

```
NODE *pop(NODE **head)
{
        NODE *temp;
        temp=*head; *head=(*head)->next; return temp;
}
```

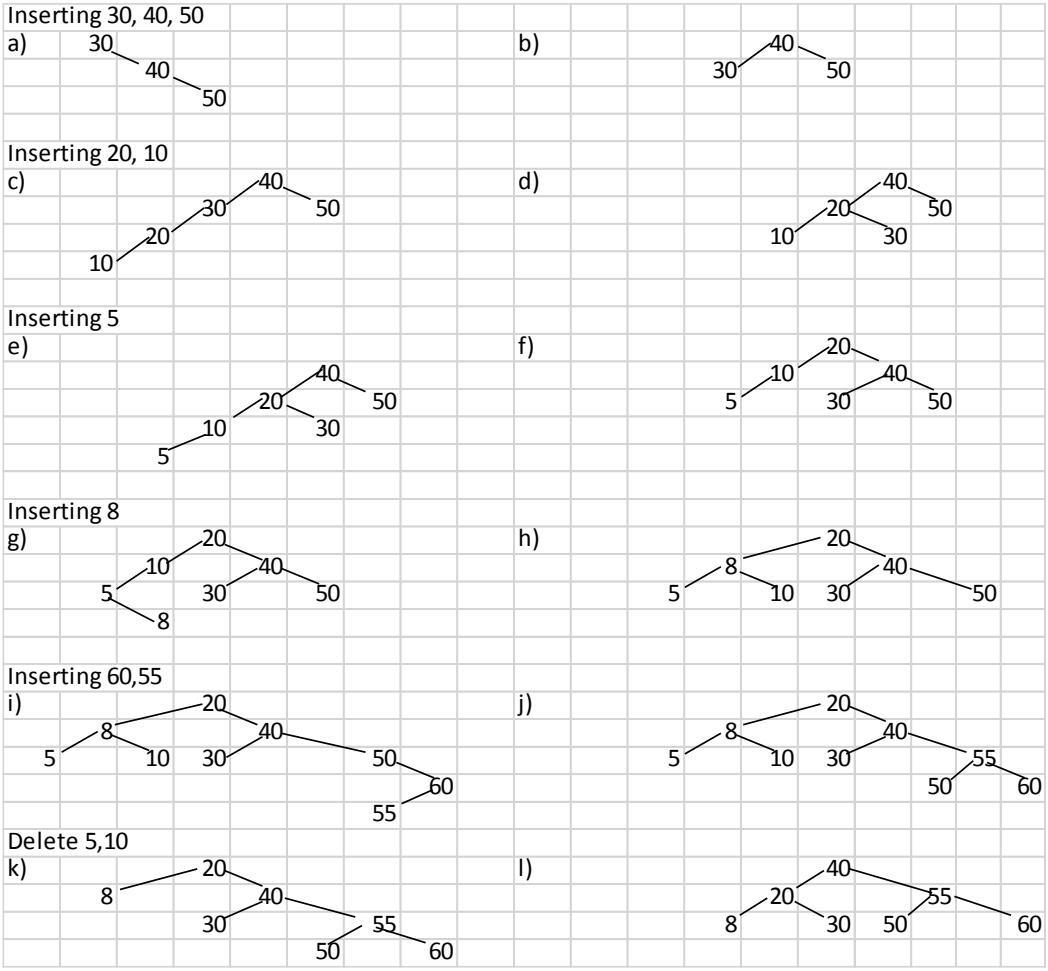
Registration Number: _____

```
NODE *enqueue(NODE *start, NODE *n)
{       n->next=NULL;
        If (start==NULL)
                return n;
        NODE *start1=start;
        while (start->next!=NULL)
                start=start->next;
        start->next=n;
        return start1;
}
```

4. a. Construct an AVL tree using the following values in the sequence given below:

30, 40, 50, 20, 10, 5,8, 60, 55

Delete nodes (in sequence) having values 5, 10 [5]



b. Write a non-recursive function to count number of non-leaf nodes of a binary search tree. The function prototype for counting the number of non-leaf nodes of a binary search tree is as below:

int countNonLeaf(NODE * root); [8]

Answer

```
int  countNonLeaf(NODE * root)
{
        NODE *p;
        int c=0;
        STACK s;
        s.top=-1;
```

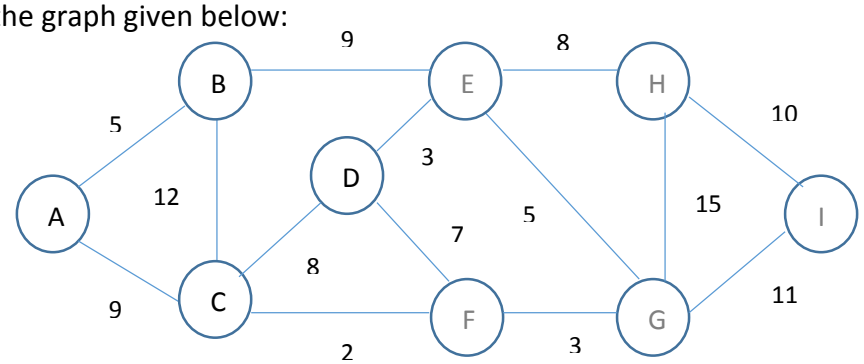
```
p=root;
        do
        {
                while(p!=NULL)
                {
                        push(&s,p);
                        p=p->left;
                }
        p=pop(&s);
        if (p->left!=NULL || p->right!=NULL)
                c++;
        p=p->right;
}while(p!=NULL || !empty(&s));
return c;}
```

c. Write a recursive function to traverse in binary search tree in postorder. [3]

Answer

void postorder(NODE *root){ if (root!=NULL) {
postorder(root->left); postorder(root->right); printf("%d ", root->info); }

5. For the graph given below:



(i) Give the breadth first traversal of the graph with starting graph node as A. Show traversal steps. [3]

(ii) Give the depth first traversal of the graph with starting graph node as A. Show traversal steps. [3]

(iii) Construct the minimum spanning tree using Kruskal Algorithm and also compute the total weight of the spanning tree.
Show the spanning tree construction steps. [3]

(iv) Construct the minimum spanning tree using Prims Algorithm and also compute the total weight of the spanning tree. Show the spanning tree construction steps. [3]

Answer

(i) BFS : one of the traversal in BFS is : A,B,C,E,D,F,H,G,I

(ii) DFS: one of the traversal in DFS is : A, C,F,G,I,H,E,D,B

(iii) Kruskal Algorithm : Sequence of edges selected: CF,FG,DE, EG, AB,EH,AC, HI
Spanning Tree weight : 45

(iv) Prims Algorithm : Sequence of edges selected: CF,FG,GE,ED,EB,BA,EH,HI
Spanning Tree weight : 45

b. Construct a B Tree of order 3 using the following values:

50, 25, 60, 70, 80, 10, 15, 40, 75, 85, 7, 8, 3, 2

Show the B tree construction steps [4]

6. a. Using the modulo-division hash function, store the following keys into a hash table of size 9.

655, 777, 45, 98, 111,891,950,72,428

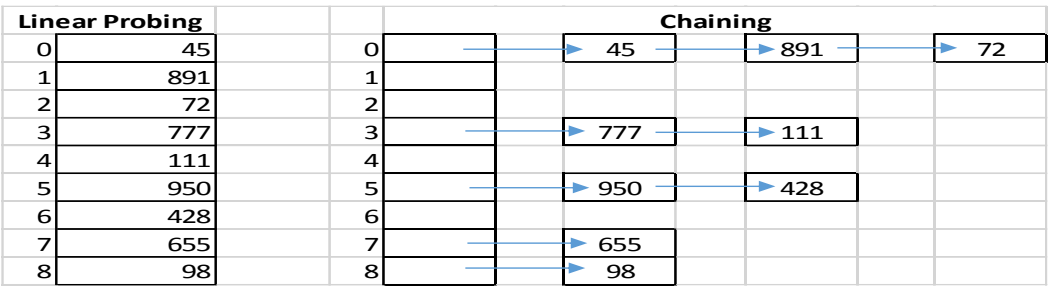
At what index positions the keys are mapped into the table if the hash collisions are resolved using

i). Linear Probing Technique [2]

ii). Separate Chaining Technique [2]

Answer

Linear Probing	
0	45
1	891
2	72
3	777
4	111
5	950
6	428
7	655
8	98



b. Write a function to merge two sorted (in ascending order) integer arrays A and B of size m and n respectively to obtain a third sorted array. [4]

Answer

```
void merge(int A[], int B[], C[], int m, int n)
{
        int i, j,k;
        for(i=0,k=0;i<m && j<n;k++)
        {
                if (A[i]<B[j])
                {
                        C[k]=A[i]; i++;
                }
                else
                {
                        C[k]=B[j];  j++;
                }
        }
        while (i<m)
        {
                        C[k]=A[i]; i++; k++;
        }
        while (k<n)
        {
                        C[k]=B[j]; j++; k++;
        }
}
```

c. Write a function to build a max-heap tree implemented using array. The function prototype for building max-heap is as below:

Heapify(int A[ ], int N, int pos)

Where A is an array of integers of size N and *pos* is the index of the tree node from where the repairing of the tree is required for max-heap. [8]

Answer

```
int heapify(int A[ ],int N, int pos)
{
        int lc,rc,l=pos;
        lc=2*pos+1; rc=2*pos+2;
        if (lc<N && A[lc]>a[l])
                l=lc;
        if (rc<N && A[rc]>a[l])
                l=rc;
        if (l!=pos)
                {
                        temp=A[pos];
                        A[pos]=a[l];
                        A[l]=temp;
                        heapify(A,N,l);
                }
}
```