

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

IAT_3 SCHEME & SOLUTION

Descriptive: 4Q*5M=20M

Multiple Choice: 5Q*2M=10M

Multiple Choice: 20Q*1M=20M

Total Marks = 50M

Descriptive: **4Q*5M=20M**

1Q. Write a C function to delete a node from binary search tree.

Solution:

C function to delete a node from binary search tree

```
struct node* deleteNode(struct node* root, int key)
```

```
{
```

```
    // base case
```

```
    if (root == NULL)
```

```
        return root;
```

```
    // If the key to be deleted
```

```
    // is smaller than the root's
```

```
    // key, then it lies in left subtree
```

```
    if (key < root->key)
```

```
        root->left = deleteNode(root->left, key);
```

```
    // If the key to be deleted
```

```
    // is greater than the root's
```

```
    // key, then it lies in right subtree
```

```
    else if (key > root->key)
```

```

root->right = deleteNode(root->right, key);

// if key is same as root's key,
// then This is the node
// to be deleted
else {
    // node with only one child or no child
    if (root->left == NULL) {
        struct node* temp = root->right;
        free(root);
        return temp;
    }
    else if (root->right == NULL) {
        struct node* temp = root->left;
        free(root);
        return temp;
    }

    // node with two children:
    // Get the inorder successor
    // (smallest in the right subtree)
    struct node* temp = minValueNode(root->right);

    // Copy the inorder
    // successor's content to this node
    root->key = temp->key;

    // Delete the inorder successor
    root->right = deleteNode(root->right, temp->key);
}
return root;
}

```

2Q. Write a Cprogram to sort a given set of n numbers using insertion sort.

Solution:

```
#include<stdio.h>
int main()
{
    int i, j, count, temp, number[25];
    printf("How many numbers u are going to enter?: ");
    scanf("%d",&count);
    printf("Enter %d elements: ", count);
    for(i=0;i<count;i++)
    scanf("%d",&number[i]);
    for(i=1;i<count;i++){
        temp=number[i];
        j=i-1;
        while((temp<number[j]&&(j>=0))
        {
            number[j+1]=number[j];
            j=j-1;
        }
        number[j+1]=temp;
    }
    printf("Order of Sorted elements: ");
    for(i=0;i<count;i++)
    printf(" %d",number[i]);
    return 0;
}
```

3Q. What is hashing? Write C program to implement Linear Probing for collision resolution.

Solution:

Hashing is an important Data Structure which is designed to use a special function called the Hash function which is used to map a given value with a particular key for faster access of elements.

Program:

```
#include<stdio.h>
#include<stdlib.h>
#define MAX 10
int f[MAX] = { 1246,1221,1235,1231,1228,1237,1218,1243,1229,1233 },HT[MAX]={0},L;
void probe(int,int);
void display();
void main()
{
int i, num;
for(i=0;i<MAX;i++)
{
L=f[i]%MAX;// find hash table entry
if(HT[L]==0) // if it empty placed key calculated position
HT[L]=f[i];
else
probe(f[i]+1,f[i]);// otherwise do linear probing
}
display();// display hash entries
}
void probe(int k,int key)
{
L=k%MAX;
if(HT[L]==0)
HT[L]=key;
else
probe(k+1,key);
```

```

}
void display()
{
int i;
printf("\nHash Table:");
for(i=0;i<MAX;i++)
{
printf("\n%d",HT[i]);
}
}
}

```

4Q. Write a c program to print all the nodes reachable from a given starting node in a digraph using BFS method.

Solution:

```

#include <stdio.h>
#include <stdlib.h>
#define size 20

void bfs(int amat[][size], int visited[], int src, int n);

void main()
{
    int n, amat[size][size], source, visited[size], i, j;
    printf("Enter the no. of cities\n");
    scanf("%d", &n);
    printf("Enter the Coef. Adjacency Matrix\n");
    for (i = 0; i < n; i++)
        for (j = 0; j < n; j++)
            scanf("%d", &amat[i][j]);
    printf("Enter Source\n");
    scanf("%d", &source);
    for (i = 0; i < n; i++)
        visited[i] = 0;
    bfs(amat, visited, source, n);
    for (i = 0; i < n; i++)
    {
        if (visited[i] == 0)
            printf("%d is not reachable\n", i);
        else
            printf("%d is reachable\n", i);
    }
}

```

```

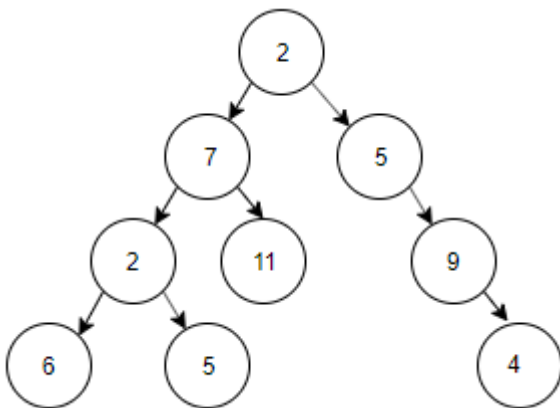
void bfs(int amat[][size], int visited[], int src, int n)
{
    int Q[size], r = 0, f = 0, u, v;
    visited[src] = 1;
    Q[r] = src;
    while (f <= r)
    {
        u = Q[f++];
        for (v = 0; v < n; v++)
        {
            if ((amat[u][v] == 1) && (visited[v] == 0))
            {
                Q[++r] = v;
                visited[v] = 1;
            }
        }
    }
}

```

Multiple Choice:

5Q*2M=10M

1Q. For the tree below, write the in-order traversal.



- a) 6, 2, 5, 7, 11, 2, 5, 9, 4
- b) 6, 5, 2, 11, 7, 4, 9, 5, 2
- c) 2, 7, 2, 6, 5, 11, 5, 9, 4
- d) 2, 7, 6, 5, 11, 2, 9, 5, 4

2Q. For the expression $a^*b/c+e/f^*g-k/h$ which of the following is the post order tree traversal?

- a) $ab^*c/ef/g^*+kh/-$
- b) $ab-c/ef/g^*+kh/^*$
- c) $ab^*c/ef/g^*+kh^*-$
- d) $ab/c^*ef/g^*+kh/-$

3Q. For the given in order and preorder traversal find the post order traversal.

Inorder: D G B A H E I C F Preorder: A B D G C E H I F

- a) G B D H I E F C A
- b) **G D B H I E F C A**
- c) G H D B I E F C A
- d) G B H I D E F C A

4Q. For the given in order and postorder traversal find the preorder traversal.

Inorder: 4 8 2 5 1 6 3 7 Post order: 8 4 5 2 6 7 3 1

- a) 8 4 5 2 6 7 3 1
- b) 8 5 4 6 2 3 7 1
- c) 5 8 4 6 2 3 7 1
- d) **1 2 4 8 5 3 6 7**

5Q. For the given in order and preorder traversal find the post order traversal.

Inorder: D B E A F C Preorder: A B D E C F

- a) D B E F C A
- b) **D E B F C A**
- c) B E D C A F
- d) B D E F A C

Multiple Choice:

20Q*1M=20M

1Q. Depth First Search is equivalent to which of the traversal in the Binary Trees?

- a) **Pre-order Traversal**
- b) Post-order Traversal
- c) Level-order Traversal
- d) In-order Traversal

2Q. The Data structure used in standard implementation of Depth First Search is?

- a) **Stack**
- b) Queue
- c) Linked List
- d) Tree

3Q. The Depth First Search traversal of a graph will result into?

- a) Linked List
- b) **Tree**
- c) Graph with back edges
- d) Array

4Q. In Depth First Search, how many times a node is visited?

- a) Once
- b) Twice
- c) **Equivalent to number of indegree of the node**
- d) Thrice

5Q. The Data structure used in standard implementation of Breadth First Search is?

- a) Stack
- b) **Queue**
- c) Linked List
- d) Tree

6Q. Breadth First Search is equivalent to which of the traversal in the Binary Trees?

- a) Pre-order Traversal
- b) Post-order Traversal
- c) **Level-order Traversal**
- d) In-order Traversal

7Q. Binary search can be used in an insertion sort algorithm to reduce the number of comparisons.

- a) **True**
- b) False

8Q. What will be the number of passes to sort the elements using insertion sort?

14, 12, 16, 6, 3, 10

- a) 6
- b) **5**

- c) 7
- d) 1

9Q. Which of the following problems occur due to linear probing?

- a) **Primary clustering**
- b) Secondary collision
- c) Separate chaining
- d) Extendible hashing

10Q. Which of the following is not a collision resolution strategy for open addressing?

- a) Linear probing
- b) Quadratic probing
- c) Double hashing
- d) **separate chaining**

11Q. The height of a BST is given as h . Consider the height of the tree as the no. of edges in the longest path from root to the leaf. The maximum no. of nodes possible in the tree is?

- a) $2^{h-1} - 1$
- b) **$2^{h+1} - 1$**
- c) $2^h + 1$
- d) $2^{h-1} + 1$

12Q. Which of the following statement about binary tree is CORRECT?

- a) Every binary tree is either complete or full
- b) Every complete binary tree is also a full binary tree
- c) **Every full binary tree is also a complete binary tree**
- d) A binary tree cannot be both complete and full

13Q. Which type of traversal of binary search tree outputs the value in sorted order?

- a) Pre-order
- b) **In-order**
- c) Post-order
- d) None

14Q. If a node having two children is to be deleted from binary search tree, it is replaced by its

- a) In-order predecessor
- b) In-order successor
- c) **Both a & b**
- d) None

15Q. A threaded binary tree is a binary tree in which every node that does not have right child has a thread to its

- a) Pre-order successor
- b) **In-order successor**
- c) In-order predecessor
- d) Post-order successor

16Q. What is a strict binary tree?

- a) Each node has exactly zero or two children
- b) **All internal nodes must have 2 children**
- c) All the leaves are at the same level
- d) Each node has exactly one or two children

17Q. What are the disadvantages of normal binary tree traversals?

- a) **there are many pointers which are null and thus useless**
- b) there is no traversal which is efficient
- c) complexity in implementing
- d) improper traversals

18Q. The leaves of an expression tree always contain?

- a) operators
- b) **operands**
- c) null
- d) expression

19Q. What does the other nodes of an expression tree(except leaves) contain?

- a) only operands
- b) **only operators**
- c) both operands and operators
- d) expression

20Q. An expression tree is created using?

- a) **postfix expression**
- b) prefix expression
- c) infix expression
- d) paranthesized expression