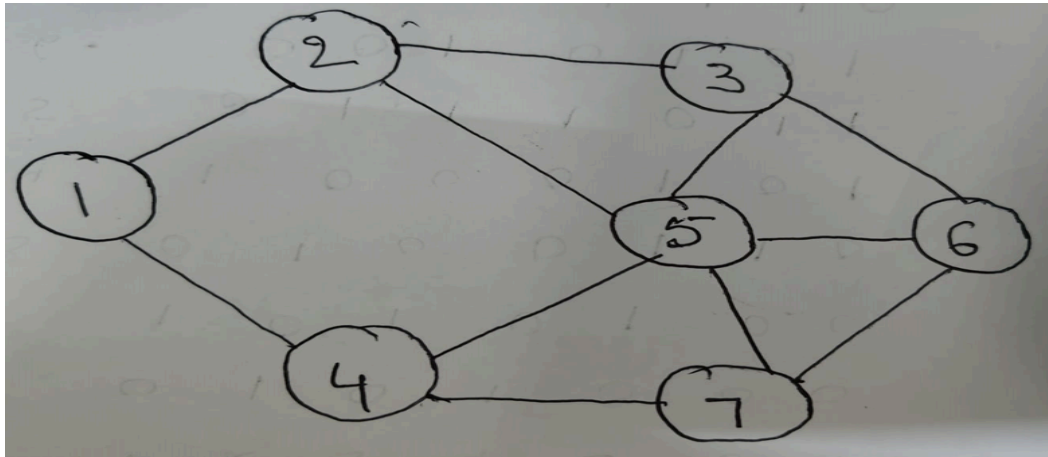


S.No	Date	IDX	PROGRAM
3	16-07-2015	3.1	Implement BFT and DFT for given graph, when graph is represented by Adjacency Matrix
		3.2	Implement BFT and DFT for given graph, when graph is represented by Adjacency Lists

Week 3 - Program 3.1

AIM : Implement BFT and DFT for given graph, when graph is represented by Adjacency Matrix.

Consider Example Given Graph



The Adjacency Matrix is

```
0 1 0 1 0 0 0
1 0 1 0 1 0 0
0 1 0 0 1 1 0
1 0 0 0 1 0 1
0 1 1 1 0 1 1
0 0 1 0 1 0 1
0 0 0 1 1 1 0
```

Adjacency Lists are

```
1 □ 2 □ 4
2 □ 1 □ 3 □ 5
3 □ 2 □ 5 □ 6
4 □ 1 □ 5 □ 7
5 □ 2 □ 3 □ 4 □ 6 □ 7
6 □ 3 □ 5 □ 7
7 □ 4 □ 5 □ 6
```

Starting from node 1

BFT : 1 2 4 3 5 7 6

DFT : 1 2 3 5 4 7 6

PROGRAM :

```
#include<stdio.h>
int a[20][20],q[20],visited[20],n,i,j,f=0,r=-1;

void dfs(int v)
{
    int k;
    visited[v]=1;
    printf("%d ",v);
    for(k=1;k<=n;k++)
    {
        if(a[v][k] && !visited[k])
            dfs(k);
    }
}

void bfs(int v)
{
    visited[v]=1;
    q[++r]=v;
    while(f!=r+1)
    {
        v=q[f++];
        printf("%d ",v);
        for(i=1;i<=n;i++)
            if(a[v][i] && !visited[i])
            {
                q[++r]=i;
                visited[i]=1;
            }
    }
}

int main()
{
    int v;
    printf("\n Enter the number of vertices:");
    scanf("%d",&n);
```

```

for(i=1;i<=n;i++)
{
    q[i]=0;
    visited[i]=0;
}
printf("\n Enter graph data in matrix form:\n");
for(i=1;i<=n;i++)
    for(j=1;j<=n;j++)
        scanf("%d",&a[i][j]);
printf("\n Enter the starting vertex:");
scanf("%d",&v);
printf(" BFT of the Graph is : \n ");
bfs(v);

for(i=1;i<=n;i++)
    visited[i]=0;

printf(" \nDFT of the Graph is : \n ");
dfs(v);
return 0;
}

```

OUTPUT:

```

Enter graph data in matrix form:
0 1 0 1 0 0 0
1 0 1 0 1 0 0
0 1 0 0 1 1 0
1 0 0 0 1 0 1
0 1 1 1 0 1 1
0 0 1 0 1 0 1
0 0 0 1 1 1 0

Enter the starting vertex:1
BFT of the Graph is :
1 2 4 3 5 7 6
DFT of the Graph is :
1 2 3 5 4 7 6

```

Week 3- Program 3.2

AIM : Implement BFT and DFT for given graph, when graph is represented by Adjacency Lists

PROGRAM:

```
#include<stdio.h>
#include<stdlib.h>
int n;
struct node
{
    int data;
    struct node *next;
};
void readgraph(struct node *ad[]);
void printgraph(struct node *ad[]);
void bfs(struct node *ad[],int start,int visited[]);
void dfs(struct node *ad[],int start,int visited[]);

int main()
{
    int i,j,k,ch,start;
    printf("Enter the Number of vertices of a Graph :");
    scanf("%d",&n);
    int visited[n];
    struct node *adj[n];
    for(i=1;i<=n;i++)
    {
        adj[i] = NULL;
    }
    readgraph(adj);
    do {
        printf("\n 1. BFS 2. DFS 3. Adjacency list 4. Exit \n Enter your choice : ");
        scanf("%d",&ch);
        switch(ch)
        {
            case 1:
                printf("Enter the starting vertex : ");
                scanf("%d",&start);
                for(int k=1;k<=n;k++)
                    visited[k]=0;
```

```

        bfs(adj,start,visited);
        break;

    case 2:
        printf("Enter the starting vertex : ");
        scanf("%d",&start);
        for(int k=1;k<=n;k++)
            visited[k]=0;
        printf("DFT : ");
        dfs(adj,start,visited);
        break;
    case 3: printgraph(adj);
        break;
    }
    }while(ch!= 4);
    return 0;
}

void readgraph(struct node *ad[])
{
    struct node *newnode;
    int i,j,k,data;
    for(i=1;i<=n;i++)
    {
        struct node *last =NULL;
        printf("\nHow many neighbours for %d : ",i);
        scanf("%d",&k);
        for(j=1;j<=k;j++)
        {
            printf("Enter the value of %d neighbour of %d : ",j,i);
            scanf("%d",&data);
            newnode = (struct node*)malloc(sizeof(struct node*));
            newnode->data = data;
            newnode->next = NULL;
            if(ad[i]==NULL)
            {
                ad[i] = newnode;
            }
            else
                last->next = newnode;
            last = newnode;
        }
    }
}

```

```

    }
}
}

```

```

void printgraph(struct node *ad[])
{
    struct node *ptr = NULL;
    int i, j;
    for(i=1;i<=n;i++)
    {
        ptr = ad[i];
        printf("\n The neighbour of %d are : ",i);
        while(ptr != NULL)
        {
            printf("%d ",ptr->data);
            ptr = ptr->next;
        }
    }
}

```

```

void bfs(struct node *ad[],int start,int visited[])
{
    int queue[n],front=-1,rear=-1;
    printf("BFT : ");
    queue[++rear] = start;
    visited[start]=1;
    front++;
    while(front<=rear)
    {
        printf("\t%d",queue[front]);
        struct node *ptr;
        ptr = ad[queue[front++]];
        while(ptr != NULL)
        {
            if(visited[ptr->data]!=1)
            {
                queue[++rear] = ptr->data;
                visited[ptr->data] =1;
            }
            ptr = ptr->next;
        }
    }
}

```

```

        }
    }
}

void dfs(struct node *ad[],int start, int visited[])
{
    visited[start] = 1;
    printf("\t%d",start);
    struct node *ptr;
    ptr = ad[start];
    while(ptr!=NULL)
    {
        if(visited[ptr->data]==0)
            dfs(ad,ptr->data,visited);
        ptr = ptr->next;
    }
}

```

OUTPUT:

Enter the Number of vertices of a Graph :7

How many neighbours for 1 : 2

Enter the value of 1 neighbour of 1 : 2

Enter the value of 2 neighbour of 1 : 4

How many neighbours for 2 : 3

Enter the value of 1 neighbour of 2 : 1

Enter the value of 2 neighbour of 2 : 3

Enter the value of 3 neighbour of 2 : 5

How many neighbours for 3 : 3

Enter the value of 1 neighbour of 3 : 2

Enter the value of 2 neighbour of 3 : 5

Enter the value of 3 neighbour of 3 : 6

How many neighbours for 4 : 3

Enter the value of 1 neighbour of 4 : 1

Enter the value of 2 neighbour of 4 : 5

Enter the value of 3 neighbour of 4 : 7

How many neighbours for 5 : 5

Enter the value of 1 neighbour of 5 : 2

Enter the value of 2 neighbour of 5 : 3

Enter the value of 3 neighbour of 5 : 4

Enter the value of 4 neighbour of 5 : 6

Enter the value of 5 neighbour of 5 : 7

How many neighbours for 6 : 3

Enter the value of 1 neighbour of 6 : 3

Enter the value of 2 neighbour of 6 : 5

Enter the value of 3 neighbour of 6 : 7

```

How many neighbours for 7 : 3
Enter the value of 1 neighbour of 7 : 4
Enter the value of 2 neighbour of 7 : 5
Enter the value of 3 neighbour of 7 : 7

1. BFS 2. DFS 3. Adjacency list 4. Exit
Enter your choice : 1
Enter the starting vertex : 1
BFT : 1      2      4      3      5      7      6
1. BFS 2. DFS 3. Adjacency list 4. Exit
Enter your choice : 2
Enter the starting vertex : 1
DFT : 1      2      3      5      4      7      6
1. BFS 2. DFS 3. Adjacency list 4. Exit
Enter your choice : 3

The neighbour of 1 are : 2 4
The neighbour of 2 are : 1 3 5
The neighbour of 3 are : 2 5 6
The neighbour of 4 are : 1 5 7
The neighbour of 5 are : 2 3 4 6 7
The neighbour of 6 are : 3 5 7
The neighbour of 7 are : 4 5 7
1. BFS 2. DFS 3. Adjacency list 4. Exit
Enter your choice : 4

```