28/2/24

WEEK-9

// BFS and DFS traversal.

```
#include <stdio.h>
#include <stdlib.h>
struct node{
    int data;
    struct node *next;
}*front=NULL, *rear=NULL;
void enqueue(int x){
  struct node *t = (struct node*) malloc (sizeof (struct node));
  if (t==NULL){
      printf ("queue is overflow");
  }
  else{
      t->data=x;
      t->next=NULL;
      if (front==NULL){
          front=rear=t;
      }
      else{
          rear->next=t;
          rear=t;
      }
  }
}
int dequeue(){
    struct node *t;
    int x=-1;
    if (front==NULL){
        printf ("queue is empty");
        return x;
    }
```

```
    else{
        t=front;
        x=t->data;
        front=front->next;
        free(t);
        return x;
    }
}

int isempt(){
    if(front==NULL){
        return 1;
    }
    return 0;
}

void bfs(int i, int visited[], int a[][20], int n){
    int u;
    printf("bfs traversal:");
    printf("%d", i);
    visited[i-1]=1;
    enqueue(i-1);
    while(!isempt()){
        u=dequeue();
        for(int v=0; v<n; v++){
            if(a[u][v]==1 && visited[v]==0){
                printf("%d", v+1);
                visited[v]=1;
                enqueue(v);
            }
        }
    }
}
```

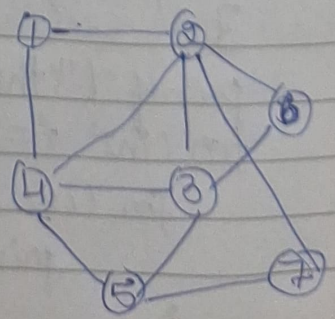
```
void dfs(int i, int visited[], int a[][20], int n){
    if(visited[i-1]==0){
        printf("%d", i);
        visited[i-1]=1;
        for(int j=0; j<n; j++){
            if(a[i-1][j]==1 && visited[j]==0){
                dfs(j+1, visited, a, n);
            }
        }
    }
}

void main(){
    int visited[20]={0};
    int a[20][20];
    int n, first;
    int count=0;
    printf("enter the number of vertices:");
    scanf("%d", &n);
    printf("enter the adjacency matrix:");
    for(int i=0; i<n; i++){
        for(int j=0; j<n; j++){
            scanf("%d", &a[i][j]);
        }
        printf
    }
    printf("the adjacency matrix:\n");
    for(int i=0; i<n; i++){
        for(int j=0; j<n; j++)
            printf("%d\t", a[i][j]);
    }
    printf("\n");
}
```

```
printf("enter the starting vertex : ");
scanf("%d", &first);
bfs(first, visited, a, n);
for(int i=0; i<20 i++){
    visited[i]=0;
}
printf("\n dfs traversal : ");
dfs(first, visited, a, n);
for(int i=0; i<n; i++){
    if(visited[i]==0){
}for(int i=0; i<n; i++){
    if(visited[i]==1){
        count++;
    }
}
if(count==n){
    printf("\n graph is connected");
}
else{
    printf("\n graph is not connected");
}
}
```

O/P enter the number of vertices : 7
enter the adjacency matrix :

0 1 0 1 0 0 0
1 0 1 1 0 1 1
0 1 0 1 1 1 0
1 1 1 0 1 0 0
0 0 1 1 0 0 1
0 1 1 0 0 0 0
0 1 0 0 1 0 0



the adjacency matrix:

0	1	0	1	0	0	0
1	0	1	1	0	1	1
0	1	0	1	1	1	0
1	1	1	0	1	0	0
0	0	1	1	0	0	1
0	1	1	0	0	0	0
0	1	0	0	1	0	0

enter the starting vertex: 4
bfs traversal: 4 1 2 3 5 6 7
dfs traversal: 4 1 2 3 5 7 6
graph is connected

Sf:1
22/2/24