NAME- Aniket Kalbhor

PRN- 12210601

ROLL- CSB-48

CODE- **8)Implement Prims Algorithm to generate MST using Adjacency Matrix.**

#include <stdio.h>

#include <stdlib.h>

#define MAX 10

void display(int G[MAX][MAX], int vertex)

{

int i, j;

for (i = 0;i < vertex;i++)

{

for (j = 0;j < vertex;j++)

printf("%d ",G[i][j]);

printf("\n");

}

}

int minDist(int dist[], int visited[], int v)

{

int min = 999, minIndex, i;

for (i = 0;i < v;i++)

if (visited[i] == 0 && dist[i] < min)

{

min = dist[i];

minIndex = i;

}

return minIndex;

}

void displayMST(int parent[], int G[MAX][MAX], int vertex)

{

int i, j;

printf("Edge \tWeight\n");

for (i = 1;i < vertex;i++)

{

printf("%d - %d \t%d\n",parent[i],i,G[i][parent[i]]);

}

}

void printArray(int arr[], int n)

{

int i;

for (i = 0;i < n;i++)

printf("%d ",arr[i]);

printf("\n");

}

void Prims(int G[MAX][MAX], int vertex)

{

int visited[MAX], dist[MAX], parent[MAX];

int i, j, u;

for (i = 0;i < vertex;i++)

{

visited[i] = 0;

dist[i] = 999;

parent[i] = -1;

}

printf("\n");

printArray(visited, vertex);

printArray(dist, vertex);

printArray(parent, vertex);

printf("\n");

dist[0] = 0;

for (i = 0;i < vertex-1;i++)

{

u = minDist(dist, visited, vertex);

visited[u] = 1;

for (j = 0;j < vertex;j++)

{

if (G[u][j] != 999 && visited[j] == 0 && G[u][j] < dist[j])

{

dist[j] = G[u][j];

parent[j] = u;

}

printf("iteration: %d\n", i+1);

printArray(visited, vertex);

printArray(dist, vertex);

printArray(parent, vertex);

printf("\n");

}

}

printf("\n");

displayMST(parent, G, vertex);

}

int main()

{

int G[MAX][MAX], vertex, e, i, j, u, v;

printf("Enter number of vertices: ");

scanf("%d",&vertex);

for (i = 0;i < vertex;i++)

for (j = 0;j < vertex;j++)

G[i][j] = 0;

printf("Enter the number of edges: ");

scanf("%d",&e);

for (i = 0;i < e;i++)

{

printf("Enter edge: ");

scanf("%d",&u);

scanf("%d",&v);

printf("Enter weight: ");

scanf("%d",&G[u][v]);

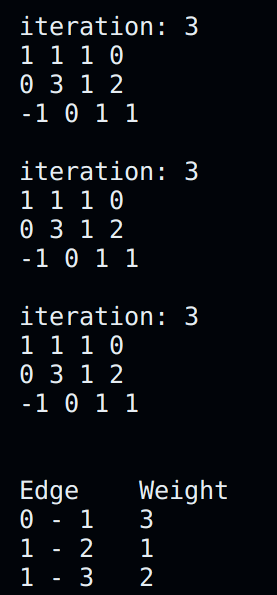
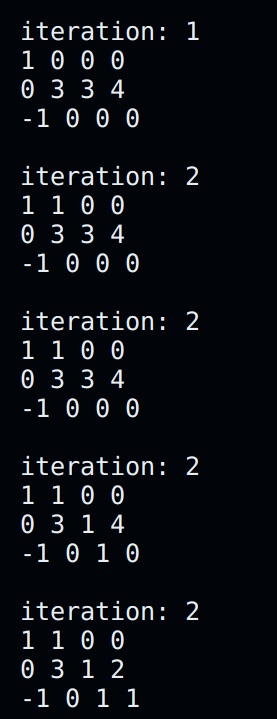
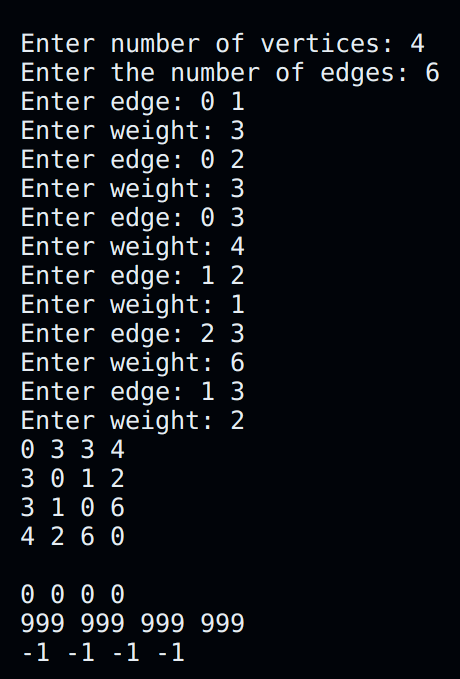
G[v][u] = G[u][v];

}

display(G, vertex);

Prims(G, vertex);

}



**Implement Prims Algorithm to generate MST using Adjacency Lists.**

#include <stdio.h>

#include <stdlib.h>

#define MAX 10

struct Node

{

int vertex;

int weight;

struct Node\* next;

};

struct Node\* a[10];

int findMinDist(int dist[10], int visited[10], int v)

{

int i, index, min = 999;

for (i = 0;i < v;i++)

{

if (visited[i] == 0 && dist[i] < min)

{

min = dist[i];

index = i;

}

}

return index;

}

void printMST(int parent[10], int v)

{

printf("Edge \tWeight\n");

for (int i = 1; i < v; i++)

if (a[i] != NULL)

printf("%d - %d \t%d \n", parent[i], i, a[i]->weight);

}

void Prims(int v)

{

int visited[10], dist[10], parent[10], u;

for (int i = 0;i < v;i++)

{

visited[i] = 0;

parent[i] = -1;

dist[i] = 999;

}

dist[0] = 0;

for(int i = 0;i < v-1;i++)

{

u = findMinDist(dist, visited, v);

visited[u] = 1;

struct Node\* p = a[u];

while (p != NULL)

{

int vert = p->vertex;

int weight = p->weight;

if (!visited[vert] && weight < dist[vert])

{

parent[vert] = u;

dist[vert] = weight;

}

p = p->next;

}

}

printMST(parent, v);

}

int main()

{

struct Node\* p;

struct Node\* newV;

int u, v, vnum, weight, edges;

int count = 0;

char ch;

printf("Enter the number of vertices:");

scanf("%d",&vnum);

printf("Enter the edges:");

scanf("%d",&edges);

for (int i = 0;i < vnum;i++)

a[i] = NULL;

do {

count++;

printf("Enter edge: ");

scanf("%d",&u);

scanf("%d",&v);

printf("Enter weight: ");

scanf("%d",&weight);

newV = (struct Node\*)malloc(sizeof(struct Node));

newV->vertex = v;

newV->weight = weight;

newV->next = NULL;

p = a[u];

if (p == NULL)

a[u] = newV;

else

{

while(p->next != NULL)

p = p->next;

p->next = newV;

}

newV = (struct Node\*)malloc(sizeof(struct Node));

newV->vertex = u;

newV->weight = weight;

newV->next = NULL;

p = a[v];

if (p == NULL)

a[v] = newV;

else

{

while(p->next != NULL)

p = p->next;

p->next = newV;

}

}while(count < edges);

printf("Adjacency List\n");

for(int i = 0;i < vnum;i++)

{

printf("Vertex %d-> ",i);

p = a[i];

while (p != NULL)

{

printf("%d,%d\t",p->vertex, p->weight);

p = p->next;

}

printf("\n");

}

Prims(vnum);

return 0;

}

