20.MINIMUM SPANNING TREE

#include <stdio.h>

#include <stdlib.h>

#define MAX\_VERTICES 20

#define infinity 999

struct edge {

int src;

int dest;

int weight;

};

struct edge edges[MAX\_VERTICES];

int compare(const void\* a, const void\* b)

{

struct edge\* edge1 = (struct edge\*)a;

struct edge\* edge2 = (struct edge\*)b;

return edge1->weight - edge2->weight;

}

int isCycle(int V, int src, int dest)

{

return 0;

}

void Kruskal(int V, int E)

{

int i;

qsort(edges, E, sizeof(struct edge), compare);

printf("Minimum Spanning Tree:\n");

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

{

int src = edges[i].src;

int dest = edges[i].dest;

int weight = edges[i].weight;

if (!isCycle(V, src, dest))

{

printf("Edge (%d-%d) Weight: %d\n", src, dest, weight);

}

}

}

int main()

{

int V, E, i;

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

scanf("%d", &V);

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

scanf("%d", &E);

printf("Enter the edges along with their weights:\n");

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

{

printf("Edge %d: ", i + 1);

scanf("%d %d %d", &edges[i].src, &edges[i].dest, &edges[i].weight);

}

Kruskal(V, E);

return 0;

}

OUTPUT

