To implement Prim's MST Algorithm using Greedy Method.

Code

#include <stdio.h>

#include <limits.h>

#define MAX\_VERTICES 100

int minKey(int key[], int mstSet[], int vertices) {

int min = INT\_MAX;

int min\_index = -1;

for (int v = 0; v < vertices; ++v) {

if (!mstSet[v] && key[v] < min) {

min = key[v];

min\_index = v;

}

}

return min\_index;

}

void primMST(int graph[MAX\_VERTICES][MAX\_VERTICES], int vertices) {

int parent[MAX\_VERTICES];

int key[MAX\_VERTICES];

int mstSet[MAX\_VERTICES];

for (int i = 0; i < vertices; ++i) {

key[i] = INT\_MAX;

mstSet[i] = 0;

}

key[0] = 0;

parent[0] = -1;

for (int count = 0; count < vertices - 1; ++count) {

int u = minKey(key, mstSet, vertices);

mstSet[u] = 1;

for (int v = 0; v < vertices; ++v) {

if (graph[u][v] && !mstSet[v] && graph[u][v] < key[v]) {

parent[v] = u;

key[v] = graph[u][v];

}

}

}

// Print the MST

printf("Edge \tWeight\n");

for (int i = 1; i < vertices; ++i) {

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

}

}

int main() {

int vertices;

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

scanf("%d", &vertices);

int graph[MAX\_VERTICES][MAX\_VERTICES];

printf("Input the adjacency matrix for the graph:\n");

for (int i = 0; i < vertices; ++i) {

for (int j = 0; j < vertices; ++j) {

scanf("%d", &graph[i][j]);

}

}

primMST(graph, vertices);

return 0;

}  
  
Output:

