#include <stdio.h>

#include <limits.h>

#define MAX 100

#define INF 99999

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

int min = INF, min\_index;

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

if (mstSet[v] == 0 && key[v] < min) {

min = key[v];

min\_index = v;

}

return min\_index;

}

void printMST(int parent[], int graph[MAX][MAX], int n) {

int totalCost = 0;

printf("Edge \tWeight\n");

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

printf("%d - %d \t%d\n", parent[i] + 1, i + 1, graph[i][parent[i]]);

totalCost += graph[i][parent[i]];

}

printf("Total cost of MST: %d\n", totalCost);

}

void primMST(int graph[MAX][MAX], int n) {

int parent[MAX];

int key[MAX];

int mstSet[MAX];

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

key[i] = INF;

mstSet[i] = 0;

}

key[0] = 0;

parent[0] = -1;

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

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

mstSet[u] = 1;

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

if (graph[u][v] && mstSet[v] == 0 && graph[u][v] < key[v]) {

parent[v] = u;

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

}

}

printMST(parent, graph, n);

}

int main() {

int graph[MAX][MAX], n;

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

scanf("%d", &n);

printf("Enter the adjacency matrix (use 0 for no edge):\n");

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

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

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

if (i != j && graph[i][j] == 0)

graph[i][j] = INF;

}

primMST(graph, n);

return 0;

}

A computer screen with white text

AI-generated content may be incorrect.