7/27/24, 12:06 AM kruskal.c++

kruskal.c++

```
#include <iostream>
 1
 2
 3
    using namespace std;
 4
 5
    const int INFI = 99;
    int adj matrix[10][10];
 6
 7
 8
    void update(int root[], int u, int v, int n) {
 9
        int temp = root[v];
10
        for (int i = 1; i <= n; i++) {</pre>
11
             if (root[i] == temp)
12
                 root[i] = root[u];
13
        }
    }
14
15
16
    void kruskal(int n) {
        int u, v, mst[n-1][2], root[n+1], cost = 0;
17
18
        for (int i = 1; i <= n; i++)</pre>
19
20
             root[i] = i;
21
        int i = 0;
22
        while (i != n - 1) {
23
             int min_edge = INFI;
24
             for (int i = 1; i <= n; i++) {</pre>
25
                 for (int j = i + 1; j <= n; j++) {</pre>
26
27
                      if (adj_matrix[i][j] < min_edge) {</pre>
28
                          min_edge = adj_matrix[i][j];
29
                          u = i;
30
                          v = j;
31
                      }
32
                 }
33
             }
34
35
             adj_matrix[u][v] = adj_matrix[v][u] = INFI;
             if (root[u] != root[v]) {
36
37
                 mst[i][0] = u;
38
                 mst[i][1] = v;
39
                 cost += min edge;
40
                 update(root, u, v, n);
41
                 i++;
42
             }
        }
43
44
45
         cout << "\nThe edges of minimum spanning tree:\n";</pre>
        for (int i = 0; i < n - 1; i++)</pre>
46
             cout << "(" << mst[i][0] << ", " << mst[i][1] << ")\n";</pre>
47
        cout << "The cost of minimum spanning tree: " << cost << endl;</pre>
48
49
    }
50
51
   int main() {
```

```
52
        int n;
53
        cout << "\nEnter the number of vertices: ";</pre>
54
        cin >> n;
55
        cout << "\nEnter the adjacency matrix: ";</pre>
56
        for (int i = 1; i <= n; i++)</pre>
57
             for (int j = 1; j <= n; j++)</pre>
                 cin >> adj_matrix[i][j];
58
59
        kruskal(n);
60
        return 0;
61 }
```