

04/24/18 11:23:28 D:\git-repos\data-structure-homework\07\e22.prim.cpp

```

1  /**-----
2   * Prim Spanning Tree Algorithm
3   *
4   * Time Consumption: E \times \log{V}
5   * Mem Consumption: linear
6   * Author: cjsoft
7   * Date: 2018/01/27
8   * -----
9   */
10 #include <queue>
11 #include <vector>
12 #include <iostream>
13 #include <cstdio>
14 #include <cstring>
15 using namespace std;
16 #define EMXN 10007
17 #define VMXN 107
18 #define E eglist
19 #define iterate(NODEN, _I) for (int _I = ehead[NODEN]; _I != -1; _I =
    eglist[_I].prev)
20 struct edge {
21     int prev, v, w;
22     edge(): prev(-1), v(0), w(0) {}
23 } eglist[EMXN];
24 int ehead[VMXN], ecur;
25 inline void einit() {
26     ecur = 0;
27     eglist[0] = edge();
28     for (int i = 1; i < EMXN; ++i)
29         eglist[i] = eglist[i - 1];
30     for (int i = 0; i < VMXN; ++i)
31         ehead[i] = -1;
32 }
33 inline void addedge(int u, int v, int w) {
34     E[ecur].v = v;
35     E[ecur].w = w;
36     E[ecur].prev = ehead[u];
37     ehead[u] = ecur++;
38 }
39 struct PII {
40     int v, dis;
41     PII(): v(0), dis(0) {}
42     PII(int v, int dis): v(v), dis(dis) {}
43     bool operator<(const PII &b) const {
44         if (dis == b.dis) return v < b.v;
45         return dis > b.dis;
46     }
47 };
48 int G[107][107], n;
49 priority_queue<PII> npq;
50 char vis[VMXN];
51 int dis[VMXN];
52 int prim(int s) {
53     int ans = 0;
54     while (!npq.empty()) npq.pop();
55     memset(dis, 0x3f, sizeof(dis));
56     memset(vis, 0, sizeof(vis));

```

```

57     dis[s] = 0;
58     npq.push(PII(s, 0));
59     PII tmp;
60     while (!npq.empty()) {
61         tmp = npq.top(), npq.pop();
62         if (vis[tmp.v] || tmp.dis > dis[tmp.v]) continue;
63         vis[tmp.v] = 1;
64         ans += dis[tmp.v];
65         for (int i = 1; i <= n; ++i) {
66             if (dis[i] > G[tmp.v][i]) {
67                 dis[i] = G[tmp.v][i];
68                 npq.push(PII(i, dis[i]));
69             }
70         }
71     }
72     return ans;
73 }
74 int main() {
75     scanf("%d", &n);
76     for (int i = 1; i <= n; ++i) {
77         for (int j = 1; j <= n; ++j) {
78             scanf("%d", &G[i][j]);
79         }
80     }
81     printf("%d\n", prim(1));
82 }
83 /**
84 root ► ... > git-repos > data-structure-homework > 07 ► g++ e22.prim.cpp
85 root ► ... > git-repos > data-structure-homework > 07 ► ./a.out ◀ master
86 4
87 0 4 9 21
88 4 0 8 17
89 9 8 0 16
90 21 17 16 0
91 28
92 root ► ... > git-repos > data-structure-homework > 07 ► ◀ master
93 */

```