

04/24/18 11:23:09 D:\git-repos\data-structure-homework\07\e22.kruskal.cpp

```
1  #include <iostream>
2  #include <cstdio>
3  #include <cstring>
4  #include <algorithm>
5  #define MXN 107
6  using namespace std;
7
8  struct edge {
9      int w, u, v;
10     edge() {
11         w = u = v = 0;
12     }
13     edge (int _a, int _b, int _c) {
14         w = _a; u = _b; v = _c;
15     }
16     bool operator<(const edge &b) const {
17         if (w < b.w) return true;
18         else if (w > b.w) return false;
19         return u < b.u;
20     }
21 } egs[MXN * MXN];
22
23 int fa[MXN], cur;
24
25 inline int getfa(int n) {
26     static int t, tmp;
27     t = n;
28     while (t != fa[t]) {
29         t = fa[t];
30     }
31     while (n != fa[n]) {
32         tmp = fa[n];
33         fa[n] = t;
34         n = tmp;
35     }
36     return t;
37 }
38
39 inline void init() {
40     for (int i = 0; i < MXN; ++i) fa[i] = i;
41 }
42
43 inline void uni(int a, int b) {
44     fa[getfa(b)] = getfa(a);
45 }
46
47 int n;
48 int main() {
49     int t;
50     cur = 0;
51     scanf("%d", &n);
52     for (int i = 1; i <= n; ++i) {
53         for (int j = 1; j <= n; ++j) {
54             scanf("%d", &t);
55             if (j > i) {
56                 egs[cur++] = edge(t, i, j);
57             }
58         }
59     }
60 }
```

```

58     }
59 }
60 sort(egs, egs + cur);
61 t = 0;
62 for (int i = 0, tt = 0; i < cur && tt < n - 1; ++i) {
63     if (getfa(egs[i].u) != getfa(egs[i].v)) {
64         uni(egs[i].u, egs[i].v);
65         ++tt;
66         t += egs[i].w;
67     }
68 }
69 printf("%d\n", t);
70 return 0;
71 }
72 /**
73 root ► ... > git-repos > data-structure-homework > 07 ► g++ e22.kruskal.cpp
74 root ► ... > git-repos > data-structure-homework > 07 ► ./a.out ◀ ? master
75 4
76 0 4 9 21
77 4 0 8 17
78 9 8 0 16
79 21 17 16 0
80 28
81 root ► ... > git-repos > data-structure-homework > 07 ► ◀ ? master
82 */

```