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

```cpp
#include <iostream>
#include <cstdio>
#include <cstring>
#include <algorithm>
#define MXN 107
using namespace std;

struct edge {
    int w, u, v;
    edge() {
        w = u = v = 0;
    }
    edge (int _a, int _b, int _c) {
        w = _a; u = _b; v = _c;
    }
    bool operator<(const edge &b) const {
        if (w < b.w) return true;
        else if (w > b.w) return false;
        return u < b.u;
    }
} egs[MXN * MXN];

int fa[MXN], cur;

inline int getfa(int n) {
    static int t, tmp;
    t = n;
    while (t != fa[t]) {
        t = fa[t];
    }
    while (n != fa[n]) {
        tmp = fa[n];
        fa[n] = t;
        n = tmp;
    }
    return t;
}

inline void init() {
    for (int i = 0; i < MXN; ++i) fa[i] = i;
}

inline void uni(int a, int b) {
    fa[getfa(b)] = getfa(a);
}
int n;
int main() {
    init();
    int t;
    cur = 0;
    scanf("%d", &n);
    for (int i = 1; i <= n; ++i) {
        for (int j = 1; j <= n; ++j) {
            scanf("%d", &t);
            if (j > i) {
                egs[cur++] = edge(t, i, j);
            }
```

```
        }
    }
    sort(egs, egs + cur);
    t = 0;
    for (int i = 0, tt = 0; i < cur && tt < n - 1; ++i) {
        if (getfa(egs[i].u) != getfa(egs[i].v)) {
            uni(egs[i].u, egs[i].v);
            ++tt;
            t += egs[i].w;
        }
    }
    printf("%d\n", t);
    return 0;
}
/**
 root ▶ … ╱ git-repos ╱ data-structure-homework ╱ 07 ▶ g++ e22.kruskal.cpp
 root ▶ … ╱ git-repos ╱ data-structure-homework ╱ 07 ▶ ./a.out      ◀  master
4
0 4 9 21
4 0 8 17
9 8 0 16
21 17 16 0
28
 root ▶ … ╱ git-repos ╱ data-structure-homework ╱ 07 ▶             ◀  master
*/
```