



《好路》题目解析

题目描述：

- 小x可以从图中选出若干组不重复的「好路」，可以确定的是两组不同的「好路」不会有重叠的边。
- 由于小x有强迫症，可以确定的是，无论怎么选「好路」，图中的每一条边一定会且只会被选在一个「好路」集合当中，且每个边都会属于一个「好路」集合。
- 每个「好路」有自己的「好值」，「好值」的计算方式为：对于「好路」中的每个顶点，都对「好值」产生贡献，贡献值为「属于该好路的边」中的最大的边权。

即 $Ans = \max(w_i, w_j)$ ，其中 w_i, w_j 表示「好路」中的两条边， Ans 表示该「好路」集合中的一部分「好值」，完整的值为 $\sum Ans$ 。

- 可以确定的是小x圈出来的每组「好路」，都可以看做一个「圈」。
- 奥对，数字9是一个奇数吧？所以这个 n 也是一个奇数呢！

小x想要知道「九曲十八弯」中，对于所有「好路」，最小的「好值」的和是多少。

解题思路：本题是一道简单题，数据范围不大。
考虑一下特殊性质， n 是一个奇数，图是一个完全图。
且对于每个点，度都是偶数。
那可以考虑排序，对于两个边，权值尽量接近才能避免浪费。
已经确定每条边都会被选定，那就没有什么太多顾虑了。
时间复杂度 $O(n^2)$ ，代码看一眼就会，非常签到。

```

#include <bits/stdc++.h>
using namespace std;
typedef long long ll;

vector<int> v[1001];

int main(int argc, char *argv[]){
    ios_base::sync_with_stdio(0);
    cin.tie(0);
    int n;
    cin >> n;
    for (int i = 1; i <= n; i++){
        for (int j = i + 1; j <= n; j++){
            int a, b, c;
            cin >> a >> b >> c;
            v[a].push_back(c);
            v[b].push_back(c);
        }
    }
    long long ans = 0;
    for (int i = 1; i <= n; i++){
        sort(v[i].begin(), v[i].end());
        for (int j = 1; j < (int) v[i].size(); j += 2){
            ans += v[i][j];
        }
    }
    cout << ans << endl;
    return 0;
}
```

《ACM-ICPC区域赛真题精讲》

课程地址



感谢各位聆听!
Thanks for Listening