

B 我在北航修光缆

21101048 尹李

题意：

给出 n 个点 m 条边的无向图，求最小生成树的边权和。

- 保证存在一条通路
 - 边权在 10^9 内，最后的答案需用 `long long`
-

思路：

求最小生成树，Kruscal 和 Prim 算法

1. Kruscal

- 按边权从小到大将边排序
 - 每次选择最小的边，并判断该边连接的端点是否在同一棵树中（在已选择的边集中），不在则加入已选择的边集
 - 维护森林的集合需要用到并查集
-

代码

并查集，需要用到路径压缩，否则会超时。

```
1 int fa[maxn];
2 int get(int x) {
3     if (fa[x] == x) {
4         return fa[x];
5     }
6     return fa[x] = get(fa[x]);
7 }
8 void merge(int u, int v) {
9     fa[get(u)] = get(v);
10 }
```

```
1 long long kruskal(int n, int m) {
2     long long sum = 0;
```

```

3     for (int i = 1; i ≤ n; i++) fa[i] = i;
4     sort(e, e + m, cmp);
5     for (int i = 0; i < m; i++) {
6         int fu = get(e[i].u);
7         int fv = get(e[i].v);
8         if (fu ≠ fv) {
9             sum += e[i].len;
10            cnt--;
11            fa[fu] = fv;
12            //merge(fu, fv);
13        }
14    }
15    return sum;
16 }

```

2. Prim

- 维护一个最小生成树的子集，子集中的点都是连通的（不同于Kruscal）
- 每次在与该子图相连的边中，选出边权最小的边，加入到子图中（类似Dijkstra）
- 可以用堆优化

代码

```

1  typedef pair<long long, int> pli;
2  long long prim(int n){
3      priority_queue<pli, vector<pli>, greater<pli>> q;
4      long long sum = 0;
5      int cnt = 0;
6      dis[1] = 0;
7      q.push(make_pair(0, 1));
8      while (!q.empty() && cnt < n) {
9          long long d = q.top().first;
10         int u = q.top().second;
11         q.pop();
12         if (vis[u]) continue;
13         cnt++;
14         sum += d;
15         vis[u] = 1;
16         for (int i = head[u]; ~i; i = e[i].next) {
17             int v = e[i].v;
18             long long w = e[i].w;
19             if (w < dis[v]) {
20                 dis[v] = w;

```

```
21         q.push(make_pair(dis[v], v));
22     }
23 }
24 }
25 return sum;
26 }
```