

```
1 #include "../bits/stdc++.h"
2 #include "../Structure/union_find.hpp"
3 #include "../graph.hpp"
4
5 // verified: http://judge.u-aizu.ac.jp/onlinejudge/review.jsp?rid=3381544
6
7 // クラスカル法による最小全域木
8 // 辺を重みが小さい順に閉路を作らないように追加していく
9 //  $O(E \log V)$ ,  $V$  := 頂点数
10 int kruskal(const Edges &es, int V)
11 {
12     UnionFind uf(V);
13     int ret = 0;
14     sort(es.rbegin(), es.rend());
15     for (const auto &e : es)
16     {
17         if (!uf.find(e.from, e.to))
18         {
19             ret += e.cost;
20             uf.unite(e.from, e.to);
21         }
22     }
23     return ret;
24 }
25
```