

```
1 #include "../bits/stdc++.h"
2 // 重み付き UnionFind
3 // verified: http://judge.u-aizu.ac.jp/onlinejudge/review.jsp?rid=3380602
4 class WeightedUnionFind
5 {
6     std::vector<int> par; // 親の番号
7     std::vector<int> ws;  // 親との重みの差
8
9 public:
10     WeightedUnionFind(int n) : par(n, -1), ws(n, 0) {}
11
12     // x の親を返す
13     int find(int x)
14     {
15         if (par[x] < 0)
16         {
17             return x;
18         }
19         else
20         {
21             int parent = find(par[x]);
22             ws[x] += ws[par[x]];
23             return par[x] = parent;
24         }
25     }
26
27     // x の重み(親との相対値)を返す
28     int weight(int x)
29     {
30         find(x);
31         return ws[x];
32     }
33
34     // 頂点 x, y を (xの重み) + w = (yの重み) となるように Unite
35     bool unite(int x, int y, int w)
36     {
37         w += weight(x);
38         w -= weight(y);
39         x = find(x);
40         y = find(y);
41
42         if (x != y)
43         {
44             if (par[y] < par[x])
45             { // 多い方が根になるようにスワップする.
46                 int tmp = x;
47                 x = y;
48                 y = tmp;
49                 w = -w;
50             }
51             par[x] += par[y];
52             par[y] = x;
53             ws[y] = w;
54             return true;
55         }
56         else
57         {
58             return false;
59         }
60     }
61
62     bool same(int x, int y)
63     {
64         return find(x) == find(y);
65     }
66
67     void init(int size)
68     {
69         for (int i = 0; i < size; i++)
70         {
71             par[i] = -1;
72             ws[i] = 0;
73         }
74     }
75 };
76
```