

```

1 #include "../bits/stdc++.h"
2 // verified: https://codeforces.com/contest/1100/submission/49830281
3 // O(|V| + |E|)
4 class Scc
5 {
6     using Graph = std::vector<std::vector<int>>>;
7     int V, K;
8     Graph G, rG;
9     std::vector<int> vs;
10    std::vector<bool> used;
11    std::vector<int> cmp;
12    void dfs(int cur)
13    {
14        used[cur] = true;
15        for (const auto &v : G[cur])
16        {
17            if (!used[v])
18                dfs(v);
19        }
20        vs.push_back(cur);
21    }
22    void rdfs(int cur, int k)
23    {
24        used[cur] = true;
25        cmp[cur] = k;
26        for (const auto &v : rG[cur])
27        {
28            if (!used[v])
29                rdfs(v, k);
30        }
31    }
32
33 public:
34    Scc(int _V) : V(_V), G(_V), rG(_V), used(_V, false), cmp(_V) {}
35    void add_edge(int from, int to)
36    {
37        G[from].push_back(to);
38        rG[to].push_back(from);
39    }
40    // SCCした後の頂点の対応関係(xor頂点数)を取得する
41    std::vector<int> scc()
42    {
43        for (int i = 0; i < V; i++)
44            used[i] = false;
45        vs.clear();
46        for (int v = 0; v < V; v++)
47        {
48            if (!used[v])
49                dfs(v);
50        }
51        for (int i = 0; i < V; i++)
52            used[i] = false;
53        int k = 0;
54        for (int i = (int)vs.size() - 1; i >= 0; i--)
55        {
56            if (!used[vs[i]])
57                rdfs(vs[i], k++);
58        }
59        // 頂点数
60        K = k;
61        // cmp[i] := 頂点i がSCC後に属する強連結成分番号
62        return cmp;
63    }
64    // 強連結成分を潰した後のグラフを得る
65    Graph buildGraph()
66    {
67        scc();
68        std::vector<std::set<int>> s(K);
69        Graph ret(K);
70        for (int v = 0; v < V; v++)
71        {
72            for (const auto to : G[v])
73            {
74                s[cmp[v]].insert(cmp[to]);
75            }
76        }
77        for (int i = 0; i < K; i++)
78        {
79            for (auto j : s[i])
80            {
81                if (i != j)
82                    ret[i].push_back(j);
83            }
84        }
85        return ret;
86    }
87 };
88

```