```cpp
class Strongly_Connected_Components {
private:
  struct node {
    int id; bool done; vi to; int from;
  };
  vector<node> nodes[2];
  int n;
  stack<int> stk;
  vvi sccs;
  vi scc_gid;
  // u means the direction
  void scc_dfs(int a, int u) {
    nodes[u][a].done = true;
    Loop(i, nodes[u][a].to.size()) {
      int b = nodes[u][a].to[i];
      if (b == nodes[u][a].from) continue;
      if (!nodes[u][b].done) {
        nodes[u][b].from = a;
        scc_dfs(b, u);
      }
    }
    if (u == 0) stk.push(a);
    else {
      sccs.back().push_back(a);
    }
    return;
  }
public:
  Strongly_Connected_Components(const vvi &lst) {
    n = lst.size();
    Loop(i, 2) nodes[i].resize(n);
    Loop(i, 2) {
      Loop(j, n) {
        nodes[i][j] = { i, false,{}, -1 };
      }
    }
    Loop(i, n) {
      Foreach(j, lst[i]) {
        nodes[0][i].to.push_back(j);
        nodes[1][j].to.push_back(i);
      }
    }
    Loop(i, n) {
      if (!nodes[0][i].done) scc_dfs(i, 0);
    }
    while (stk.size()) {
      int a = stk.top(); stk.pop();
      if (!nodes[1][a].done) {
        sccs.push_back({});
        scc_dfs(a, 1);
        sort(sccs.back().begin(), sccs.back().end());
      }
    }
    return;
  }
  // already in topological order
  vvi get_sccs() {
    return sccs;
  }
  vi get_scc_gid() {
    if (scc_gid.empty()) {
      scc_gid.resize(n);
      Loop(i, sccs.size()) {
        Loop(j, sccs[i].size()) {
          scc_gid[sccs[i][j]] = i;
        }
      }
    }
    return scc_gid;
  }
};
```