```cpp
class Finding_Bridges {
private:
  struct node {
    int id; bool done; vi to; int from; int pre; int low;
  };
  vector<node> nodes;
  int n, m;
  int ord;
  vector<P> result;
  void lowlink_dfs(int a) {
    nodes[a].done = true;
    nodes[a].pre = nodes[a].low = ord;
    ord++;
    Loop(i, nodes[a].to.size()) {
      int b = nodes[a].to[i];
      if (b == nodes[a].from) continue;
      if (!nodes[b].done) {
        nodes[b].from = a;
        lowlink_dfs(b);
        nodes[a].low = min(nodes[a].low, nodes[b].low);
        if (nodes[a].pre < nodes[b].low) {
          if (a < b) result.push_back({ a,b });
          else result.push_back({ b,a });
        }
      }
      else {
        nodes[a].low = min(nodes[a].low, nodes[b].pre);
      }
    }
    return;
  }
public:
  Finding_Bridges(const vvi &lst) {
    n = lst.size();
    nodes.resize(n);
    Loop(i, n) nodes[i] = { i, false,{}, -1, -1, -1 };
    Loop(i, n) {
      Foreach(j, lst[i]) {
        nodes[i].to.push_back(j);
      }
    }
    ord = 0;
    Loop(i, nodes.size()) {
      if (!nodes[i].done) lowlink_dfs(i);
    }
    sort(result.begin(), result.end());
  }
  vector<P> get_bridges() {
    return result;
  }
};
```