```cpp
#include "../bits/stdc++.h"
// 関節点 (= 取り除くと連結成分が増加する頂点) 列挙
// 根の場合, 次数が2以上なら関節点
// otherwise, 橋列挙での low > order を low >= order にすればよい
// verified: http://judge.u-aizu.ac.jp/onlinejudge/review.jsp?rid=3382055
class ArticulationPoint
{
    using Graph = std::vector<std::vector<int>>;
    using P = std::pair<int, int>;
    std::vector<int> order;
    int next = 0;
    Graph G;
    std::vector<int> ps;

    void add(int v)
    {
        ps.push_back(v);
    }
    int dfs(int cur, int pre)
    {
        int res = order[cur] = next++;
        bool isArticulation = false;
        int d = 0;
        for (const auto &to : G[cur])
        {
            if (to == pre)
                continue;
            if (order[to] >= 0)
                res = std::min(res, order[to]);
            else
            {
                int low = dfs(to, cur);
                if (pre >= 0 && low >= order[cur])
                    isArticulation = true;
                res = std::min(res, low);
                d++;
            }
        }
        if (pre < 0 && d >= 2)
            isArticulation = true;
        if (isArticulation)
            add(cur);
        return res;
    }

  public:
    ArticulationPoint(int _v) : G(_v, std::vector<int>()), order(_v, -1) {}
    void addEdge(int s, int t)
    {
        G[s].push_back(t);
    }
    std::vector<int> points()
    {
        dfs(0, -1);
        sort(ps.begin(),ps.end());
        return ps;
    }
};

```