

```

1 #include "../bits/stdc++.h"
2
3 // 最大流(Dinic)
4 // O(EV^2) (<=> Ford : O(VE))
5 // verified: http://judge.u-aizu.ac.jp/onlinejudge/review.jsp?rid=3633135
6 class Dinic
7 {
8     using type = int;
9     const int INF = 1e9;
10    struct Edge
11    {
12        // 行き先, 容量, 逆辺
13        int to;
14        type cap;
15        int rev;
16        Edge(int _t, type _c, int _r) : to(_t), cap(_c), rev(_r) {}
17    };
18
19    using Graph = std::vector<std::vector<Edge>>;
20    int V;
21    Graph G;
22    // s からの最短距離
23    std::vector<int> level;
24    // 訪問済みか
25    std::vector<int> iter;
26
27    void bfs(int s)
28    {
29        fill(level.begin(), level.end(), -1);
30        std::queue<int> que;
31        level[s] = 0;
32        que.push(s);
33        while (!que.empty())
34        {
35            int v = que.front();
36            que.pop();
37            for (const auto &e : G[v])
38            {
39                if (e.cap > 0 && level[e.to] < 0)
40                {
41                    level[e.to] = level[v] + 1;
42                    que.push(e.to);
43                }
44            }
45        }
46    }
47    // 増加パスを探す
48    type dfs(int v, int t, type f)
49    {
50        if (v == t)
51            return f;
52        for (int &i = iter[v]; i < (int)G[v].size(); i++)
53        {
54            Edge &e = G[v][i];
55            if (e.cap > 0 && level[v] < level[e.to])
56            {
57                type d = dfs(e.to, t, std::min(f, e.cap));
58                if (d > 0)
59                {
60                    e.cap -= d;
61                    G[e.to][e.rev].cap += d;
62                    return d;
63                }
64            }
65        }
66        return 0;
67    }
68
69    public:
70    Dinic(int _v) : V(_v), G(_v), level(_v), iter(_v) {}
71    void add(int from, int to, type cap)
72    {
73        G[from].push_back(Edge{to, cap, (int)G[to].size()});
74        G[to].push_back(Edge{from, 0, (int)G[from].size() - 1});
75    }
76    type maxFlow(int s, int t)
77    {
78        type ret = 0;
79        while (true)
80        {
81            bfs(s);
82            if (level[t] < 0)
83                return ret;
84            fill(iter.begin(), iter.end(), 0);
85            type f;
86            while ((f = dfs(s, t, INF)) > 0)
87            {
88                ret += f;
89            }
90        }
91    }
92 };
93
94 /*
95 https://ei1333.github.io/algorithm/dinic.html

```

```
96 http://yosupo.hatenablog.com/entry/2015/03/31/134336
97 燃やす埋める
98 以下のような制約問題は最小カットに帰着できる
99 - 各頂点に  $\{0, 1\}$  を割り当て
100 - 頂点  $x_i$  に 0, 頂点  $y_i$  に 1 を割り当てたとき,  $z_i$  の損失
101 - 損失の和の最小値を求めよ
102
103 ->  $x_i$  から  $y_i$  に  $z_i$  の辺を張って最大流(=最小カット)が求める値に等しい
104
105  $x_i = 0$ ,  $z_i$  損失 ->  $(x_i, T, z_i)$ 
106  $x_i = 1$ ,  $z_i$  損失 ->  $(S, x_i, z_i)$  etc...
107 */
108
```