

```

1 #include "../bits/stdc++.h"
2
3 // 最大流(Ford Fulkerson)
4 // O(FE)
5 // verified: http://judge.u-aizu.ac.jp/onlinejudge/review.jsp?rid=3633122
6 class FordFulkerson
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    // dfs で訪問済みか
23    std::vector<bool> usd;
24
25    // 増加パスを dfs で探す
26    type dfs(int v, int t, type f)
27    {
28        if (v == t)
29            return f;
30        usd[v] = true;
31        for (auto &e : G[v])
32        {
33            if (!usd[e.to] && e.cap > 0)
34            {
35                auto d = dfs(e.to, t, std::min(f, e.cap));
36                if (d > 0)
37                {
38                    e.cap -= d;
39                    G[e.to][e.rev].cap += d;
40                    return d;
41                }
42            }
43        }
44        return 0;
45    }
46
47 public:
48     FordFulkerson(int _v) : V(_v), G(_v), usd(_v) {}
49     void add(int from, int to, type cap)
50     {
51         G[from].push_back(Edge{to, cap, (int)G[to].size()});
52         G[to].push_back(Edge{from, 0, (int)G[from].size() - 1});
53     }
54     // s->t の最大流を求める
55     type maxFlow(int s, int t)
56     {
57         type ret = 0;
58         while (true)
59         {
60             for (int i = 0; i < (int)G.size(); i++)
61                 usd[i] = false;
62             type f = dfs(s, t, INF);
63             if (f == 0)
64                 return ret;
65             ret += f;
66         }
67     }
68 };
69

```