UVa1658

题意：给定v（3<=v<=1000)个节点，e(3<=e<=10000)条边的又向加权图，求1->v的两条不相交的路径，使得权和最小。

|  |
| --- |
| #include <cstdio>  #include <cstring>  #include <queue>  #include <vector>  using namespace std;  #define rep(i,a,n) for(int i = a; i < n; i++)  #define repe(i,a,n) for(int i = a; i <= n; i++)  #define clc(a,b) memset(a,b,sizeof(a))  #define MAXN 2010  #define INF 0x3f3f3f3f  typedef long long LL;  struct MCMF{  int n,m;  struct Edge{  int from, to, cap, flow, cost;  Edge(int a, int b, int c, int d, int e){  from = a, to = b, cap = c, flow = d, cost = e;  }  };  vector<Edge> edge;  vector<int> g[MAXN];  bool inq[MAXN];  int d[MAXN]/\*spfa\*/, p[MAXN]/\*上一条弧\*/, a[MAXN]/\*可改进量\*/;  void init(int n){  this->n = n;  repe(i,1,n) g[i].clear();  edge.clear();  }  void add\_edge(int from, int to, int cap, int cost)  {  edge.push\_back(Edge(from,to,cap,0,cost));  edge.push\_back(Edge(to,from,0,0,-cost));  m = edge.size();  g[from].push\_back(m-2);  g[to].push\_back(m-1);  }  bool spfa(int s, int t, int& flow, LL& cost)  {  clc(d,0x3f);  clc(inq,0);  d[s] = 0, inq[s] = true, p[s] = 0, a[s] = INF;  queue<int> q;  q.push(s);  while(!q.empty())  {  int u = q.front();q.pop();  inq[u] = false;  int sz = g[u].size();  rep(i,0,sz)  {  Edge& e = edge[g[u][i]];  if(e.cap > e.flow && d[e.to] > d[u]+e.cost)  {  d[e.to] = d[u]+e.cost;  p[e.to] = g[u][i];  a[e.to] = min(a[u], e.cap-e.flow);  if(!inq[e.to]) q.push(e.to), inq[e.to] = true;  }  }  }  if(INF == d[t]) return false;  flow += a[t];  cost += (LL)d[t]\*(LL)a[t];  for(int u = t; u != s; u = edge[p[u]].from)  {  edge[p[u]].flow += a[t];  edge[p[u]^1].flow -= a[t];  }  return true;  }  //需要保证初始网络没有负圈，返回最大流量，cost才是最小花费  int mincostmaxflow(int s, int t, LL & cost)  {  int flow = 0;  cost = 0;  while(spfa(s,t,flow,cost));  return flow;  }  }mcmf;    int main()  {  #ifdef SHY  freopen("e:\\1.txt","r",stdin);  #endif  int n,m;  while(~scanf("%d %d%\*c", &n, &m))  {  int a,b,c;  mcmf.init(n<<1);  rep(i,2,n) mcmf.add\_edge(i,i+n,1,0);  mcmf.add\_edge(1,n+1,2,0);  mcmf.add\_edge(n,n<<1,2,0);  rep(i,0,m)  {  scanf("%d %d %d%\*c", &a, &b, &c);  mcmf.add\_edge(a+n,b,1,c);  }  LL ans;  mcmf.mincostmaxflow(1,n<<1,ans);  printf("%lld\n", ans);  }  return 0;  } |