========== DINIC() ==========

// O(n^2\*m)

// all con[]==1 O(min(n^(2/3),m(1/2))\*m)

// 二分图匹配 O(n^(0.5) \* m)

#define INF 0x3f3f3f3f

int m0 = 1, S, T;

int head[MaxNode],u[MaxEdge],v[MaxEdge],NEXT[MaxEdge],con[MaxEdge];

int Q[MaxQue],dis[MaxNode],cur[MaxNode],vis[MaxNode];

int Min(int a,int b){return a<b?a:b;}

void Clear() {

// include <cstring>

// memset(head u v NEXT con Q cur) to zero;

}

void Add(int a,int b,int c) {

v[++m0]=b;NEXT[m0]=head[a];head[a]=m0;con[m0]=c;

v[++m0]=a;NEXT[m0]=head[b];head[b]=m0;con[m0]=0;

}

int bfs() {

int Qhead=0,Qtail=0;

memset(vis,0,sizeof(vis));

memset(dis,127,sizeof(dis));

dis[S]=0;vis[S]=1;Q[++Qtail]=S;

while(Qhead<Qtail) {

++Qhead;

for(int i=head[Q[Qhead]];i;i=NEXT[i])

if(!vis[v[i]] && con[i]) {

vis[v[i]]=1;

dis[v[i]]=dis[Q[Qhead]]+1;

Q[++Qtail]=v[i];

}

}

return vis[T];

}

int dfs(int now,int lim) {

if (now==T || !lim) return lim;

int flow=0, f;

for (int& i=cur[now];i;i=NEXT[i]) {

if (dis[v[i]]>dis[now] && con[i])

if ((f=dfs(v[i],Min(lim-flow,con[i]))) > 0){

flow+=f;

con[i]-=f;

con[i^1]+=f;

if (flow==lim) break;

}

}

return flow;

}

int DINIC() {

int flow=0;

while(bfs()) {

memcpy(cur,head,sizeof(head));

flow+=dfs(S,INF);

}

return flow;

}

// Ans = DINIC();

// ========== 费用流 ==========

最小费用最大流m0=1; void AddEdge(int a,int b,int c,int d)

{ v[++m0]=b;u[m0]=a;con[m0]=c;cost[m0]=d;prep[m0]=head[a];head[a]=m0;

v[++m0]=a;u[m0]=b;con[m0]=0;cost[m0]=-d;prep[m0]=head[b];head[b]=m0;}

bool spfa()

{ memset(Dis,127,sizeof(Dis)); memset(vis,0,sizeof(vis));

Dis[S]=0; vis[S]=1; Que[Qhead=Qtail=1]=S;

while(Qhead<=Qtail){

for(int

i=head[Que[Qhead]];i;i=prep[i]) if(con[i]&&Dis[v[i]]>Dis[Que[Qhead]]+cost[i]){

Dis[v[i]]=Dis[Que[Qhead]]+cost[i];

path[v[i]]=i;

if(!vis[v[i]])

vis[Que[++Qtail]=v[i]] = 1;

}

vis[Que[Qhead]]=0; ++Qhead;

} return Dis[T]<2100000000; }

void CostFlow()

{ int x; Ans=0;

memset(path,0,sizeof(path));

while(spfa())

{ int f=INF;

for(x=T;x!=S;x=u[path[x]]) f=Min(f,con[path[x]]);

for(x=T;x!=S;x=u[path[x]])

{con[path[x]]-=f; con[path[x]^1]+=f; }

Ans+=Dis[T]\*(long long)f;

}}

S=1;T=2; CostFlow(); printf("%d\n",Ans);

// 最短路

// ========== SPFA ==========

vector<int> G[MaxN];

int tot=0,

v[MaxM],NEXT[MaxM],w[MaxM],head[MaxN]; //memset

int Dis[MaxN],Que[MaxQ];

bool vis[MaxN];

void AddEdge(int a,int b,int c)

{v[++tot]=b;w[tot]=c;NEXT[tot]=head[a];

head[a]=tot;}

void Spfa(int Source) {

memset(Dis,63,sizeof(Dis));

memset(vis, 0, sizeof(vis));

int Qhead=1, Qtail=1;

Dis[Que[1]=Source]=0;

while(Qhead<=Qtail) {

for(int i=head[Que[Qhead]];i;i=NEXT[i])

if(Dis[Que[Qhead]]+w[i]<Dis[v[i]]) {

Dis[v[i]]=Dis[Que[Qhead]]+w[i];

if(!vis[v[i]]) {

vis[v[i]]=1;

Que[++Qtail]=v[i];

}

}

vis[Que[Qhead]]=0; ++Qhead;

}

}

// S = ???; // Spfa();