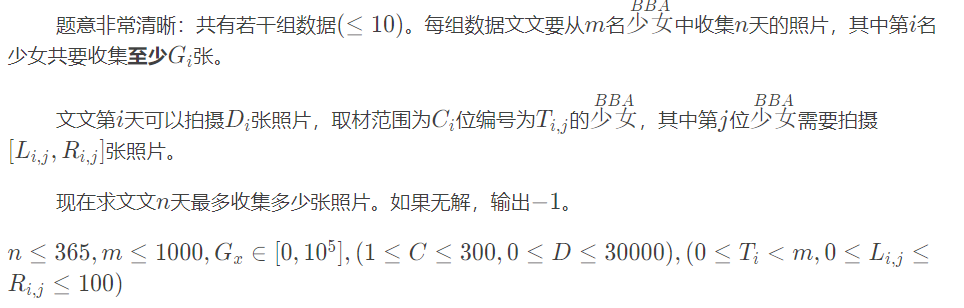
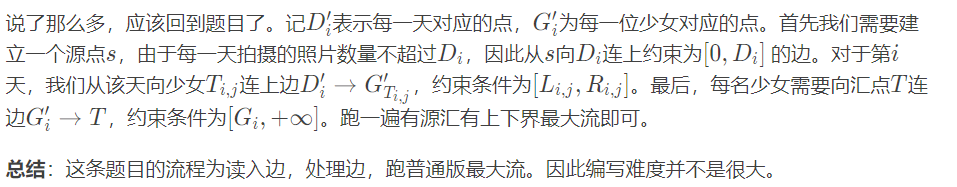


例题：luogu 5192 Zoj3229 Shoot the Bullet|东方文花帖|【模板】有源汇上下界最大流





下面是有源汇上下界最大流的代码，稍微改一下便可以对应：

无源汇上下界可行流 有源汇上下界可行流 有源汇上下界最小流

#include<bits/stdc++.h>

#define ll long long

#define inf 1e18

#define mn 1000005

using namespace std;

ll d[mn],h[mn],gap[mn],in[mn],out[mn],

cur[mn],sum,s1,t1,s2,t2,

cnt=1,ans=0,n,m,s,t,maxflow;//注意这里的cnt是从1开始的，边序号从2开始，这样（2,3）号边才是一组

struct w{ll next,to,z;}a[mn];//注意要开边数的两倍，网络流都是双向边

void add(ll x,ll y,ll z)

{

a[++cnt]=(w){h[x],y,z};h[x]=cnt;

a[++cnt]=(w){h[y],x,0};h[y]=cnt;

}

void bfs()

{ll i,x,y;

for(i=1;i<=n;i++){d[i]=-1;gap[i]=0;}

d[t]=0;gap[0]=1;

queue<ll>q;

q.push(t);

while(q.empty()==0)

{

x=q.front();

q.pop();

for(i=h[x];i;i=a[i].next)

{

y=a[i].to;

if(d[y]!=-1)continue;

q.push(y);

d[y]=d[x]+1;

gap[d[y]]++;

}

}

}

ll dfs(ll x,ll flow)

{ll y;

if(x==t){maxflow+=flow;return flow;}

ll used=0;

for(ll &i=cur[x];i;i=a[i].next)//cur[]即为弧优化

{

y=a[i].to;

if(a[i].z&&d[y]+1==d[x])

{

ll mi=dfs(y,min(a[i].z,flow-used));

if(mi){a[i].z-=mi;a[i^1].z+=mi;used+=mi;}

if(used==flow)return used;

}

}

if(--gap[d[x]]==0)d[s]=n+1;

d[x]++;gap[d[x]]++;

return used;

}

ll isap()

{ll i;

maxflow=0;

bfs();

while(d[s]<n){for(i=1;i<=n;i++)cur[i]=h[i];dfs(s,inf);}

return maxflow;

}

int main()

{

ll x,y,z,i,j,k;

char ch;

while(cin>>n>>m)

{

sum=0;cnt=1;

memset(h,0,sizeof(h));

memset(in,0,sizeof(in));//in[]表示节点的流入之和

memset(out,0,sizeof(out));//out[]表示节点的流出之和

s1=n+m+1;t1=n+m+2;s2=n+m+3;t2=n+m+4;//s1,t1为原来的源汇，s2,t2为增加的源汇

s=s2;t=t2;//s,t是在isap里面的源汇点

for(i=1;i<=m;i++)

{

scanf("%lld",&x);

in[t1]+=x;out[n+i]+=x;//每次建边都要加一次in[]和out[]

add(n+i,t1,inf-x);//建边的流量为上界减去下界

}

for(i=1;i<=n;i++)

{

scanf("%lld%lld",&k,&x);

add(s1,i,x);

for(j=1;j<=k;j++)

{

scanf("%lld%lld%lld",&x,&y,&z);

x++;//这个是因为原题节点是从0开始的

in[n+x]+=y;out[i]+=y;//每次建边都要加一次in[]和out[]

add(i,n+x,z-y);//建边的流量为上界减去下界

}

}

for(i=1;i<=n+m+2;i++)//这一步就是在补齐原图的流量

if(in[i]>out[i]){add(s,i,in[i]-out[i]);sum+=in[i]-out[i];}

else add(i,t,out[i]-in[i]);

add(t1,s1,inf);//有源汇上下界需要加上这一行，无源汇就不用加了

n=n+m+4;//因为跑最大流的时候里面要用到总结点数n

if(isap()!=sum)//最大流等于sum，才有可行流，如果只是判断可行流，就可以到此为止了

{

printf("-1\n\n");

continue;

}

//下面是在求上下界最大流

ans=a[cnt].z;

a[cnt].z=a[cnt^1].z=0;

s=s1;t=t1;//如果是要求上下界最小流，就把s=t1，t=s1，下面变为ans-=isap()即可

ans+=isap();

printf("%lld\n\n",ans);

}

return 0;

}