[解题报告]F Gems

[Source] http://202.114.18.202:8080/judge/contest/contest/view.action?cid=6147#problem/F

[Description]

Alsomagic有n种形状m种颜色的宝石，他要送给GF一些。Alsomagic不希望留在手中的宝石某种形状的太多，GF不希望得到的宝石某种颜色的太多。Alsomagic可以将某两种宝石互相转换，如可以将A形状B颜色，转换为C形状D颜色，反之亦可。每种转换只能使用一次（注意，是一次即一个方向，而不是一个宝石）。问能不能把这些宝石进行合适的转换和分配，满足两个人的要求。

[Solution]

这是一道网络流建图题，约束关系有宝石的个数，每种形状和颜色最多的个数，还有宝石不同种类间的相互转化。建立m*n个结点代表不同种的宝石，m个结点代表m种形状，n个结点代表n种颜色，A点代表Alsomagic，B点代表GF，S点源，T点汇。

S点连每一种宝石，容量为该种宝石的个数，i形状j颜色的宝石连第i种形状，第j种颜色，容量为无穷，可以转化的宝石间连边（双向），容量无穷。每种形状连A，容量为该种形状的宝石最多的个数，每种颜色连B，容量为该种颜色的宝石最多的个数。A和B连T，容量为无穷。

从S到T求一遍最大流，如果流量等于宝石的总数即可以满足要求，否则不能满足。

[Code]

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>

using namespace std;
const int inf=99999999;
const int N=50000;
struct edge
{
	int x,y,nxt;
	int c;
}bf[N];
int ne,head[N],cur[N],ps[N],dep[N];
```

```
void add(int x,int y,int c1,int c2)
{
	bf[ne].x=x; bf[ne].y=y; bf[ne].c=c1;
	bf[ne].nxt=head[x]; head[x]=ne++;
	bf[ne].x=y; bf[ne].y=x; bf[ne].c=c2;
	bf[ne].nxt=head[y]; head[y]=ne++;
}

int flow(int n,int s,int t)
{
	int tr,res=0;
	int i,j,k,f,r,top;
	while (1)
	{
		memset(dep,-1,sizeof(dep));
		for (f=dep[ps[0]=s]=0,r=1; f!=r; )
		  for (i=ps[f++],j=head[i]; j; j=bf[j].nxt)
		    {
			if (bf[j].c>0 && dep[k=bf[j].y]==-1)
			{
				dep[k]=dep[i]+1; ps[r++]=k;
				if (k==t) {f=r; break;}
			}
		    }
		if (dep[t]==-1) break;
		memcpy(cur,head,n*sizeof(int));
		for (i=s,top=0; ; )
		{
			if (i==t)
			{
				for (k=0,tr=inf; k<top; ++k)
				  if (bf[ps[k]].c<tr)
				    tr=bf[ps[f=k]].c;
				for (k=0; k<top; ++k)
				  bf[ps[k]].c-=tr,bf[ps[k]^1].c+=tr;
				res+=tr; i=bf[ps[top=f]].x;
			}
			for (j=cur[i]; cur[i]; j=cur[i]=bf[cur[i]].nxt)
			  if (bf[j].c && dep[i]+1==dep[bf[j].y]) break;
			if (cur[i])
			{
				ps[top++]=cur[i];
				i=bf[cur[i]].y;
```

```
                }
                else
                {
                        if (top==0) break;
                        dep[i]=-1;
                        i=bf[ps[--top]].x;
                }
        }
    }
    return res;
}

int main()
{
    int o,m,n,i,j,x,y,p,q,k,s,t;
    scanf("%d",&o);
    while (o--)
    {
        scanf("%d%d",&n,&m);
        memset(head,0,sizeof(head)); ne=2;
        int tol=0;
        s=n*m+n+m+2;
        t=n*m+n+m+3;
        for (i=0; i<n; i++)
          for (j=0; j<m; j++)
            {
                scanf("%d",&x); tol+=x;
                add(s,i*m+j,x,0);
                add(i*m+j,n*m+i,inf,0);
                add(i*m+j,n*m+n+j,inf,0);
            }
        scanf("%d",&k);
        for (i=1; i<=k; i++)
        {
            scanf("%d%d",&x,&y);
            p=x*m+y;
            scanf("%d%d",&x,&y);
            q=x*m+y;
            add(p,q,inf,inf);
        }
        for (i=0; i<n; i++)
        {
            scanf("%d",&x);
            add(n*m+i,n*m+n+m,x,0);
```

```
            }
            for (i=0; i<m; i++)
            {
                scanf("%d",&x);
                add(n*m+n+i,n*m+n+m+1,x,0);
            }
            add(n*m+n+m,t,inf,0);
            add(n*m+n+m+1,t,inf,0);
            int sum=flow(n*m+n+m+4,s,t);
            if (sum==tol) printf("Yes\n"); else printf("No\n");
        }
}
```