

t1 Guard

题目描述

过年了，小偷们也都忙着置办年货了，为了保护自己公司的财产安全，小A打算在公司里配备一些警卫。小A的公司由 m 栋大楼构成，其中第 i 栋楼有 K_i 层。这 m 栋楼房顺时针构成了一个环形，并且相邻两栋楼的两个楼层之间可能建造有连廊。然而小A是非常抠门的，他既希望部署尽可能多的警卫，又希望不要有两个警卫在同一个楼的相同楼层，或者他们所在的楼层有连廊连接。现在小A找到了你为他安排一下警卫们的位置，看看最多能安排多少个警卫，事成之后.....你懂的。

输入格式

第一行一个整数 m ，表示大楼的个数。

接下来若干个整数三元组 A, B, C ，表示第 C 栋楼的第 A 层和它顺时针下一栋楼的第 B 层之间有一个连廊。

（第 i 栋楼的楼层数等于上边三元组中出现过的第 i 栋楼的最大楼层数）

输出格式

一行一个整数 t ，表示最多安排的警卫人数。

样例输入

```
3
1 1 1
1 2 1
1 1 2
```

样例输出

```
2
```

数据规模与约定

对于30%的数据， $\sum K_i \leq 20$ 。

对于另外30%的数据， $K_i \leq 5$ 。

对于100%的数据， $10 \leq m \leq 50$ ， $\sum_{i=1}^m K_i \leq 100$ 。

题解&反思

因为某些投机做过的原因导致A了。首先对于偶数栋楼，可以直接建二分图跑dinic。对于奇数，我一开始想建的是新加一栋然后用这栋楼代替原来的第 n 栋和1连边，并且原来的 n 和新加的 n 每个房间都连边表示只能选一个，然而这是错的。考虑一个暴力的做法：根据鸽巢原理我也不知道为啥一定会有一栋房间少于10的楼，我们枚举这栋楼的状态，并且把与这栋楼相连的边全部删掉（注意与它相邻的两栋楼与它连边的房间，如果当前枚举状态下这栋楼的 i 房间选，那么与其相连的房间都应不选）。然后把编号大于这栋楼的楼的编号全部建议，于是这又是一个二分图了。

然而即使是做过也在调试上花了一个多小时orz，代码能力要提高啊啊！

```

#include<iostream>
#include<cstdio>
#include<cstring>
#include<queue>
using namespace std;
const int N=100005,inf=1e9;
int n,m,c[105],s,t,cnt=1,h[N],le[N],sum[105],ans,an;
bool del[105];
struct qwe
{
    int ne,to,va;
}e[N<<2];
struct qw
{
    int a,b,c;
}a[N];
void add(int u,int v,int w)
{
    cnt++;
    e[cnt].ne=h[u];
    e[cnt].to=v;
    e[cnt].va=w;
    h[u]=cnt;
}
void ins(int u,int v,int w)
{
    //cout<<u<<" "<<v<<" "<<w<<endl;
    add(u,v,w);
    add(v,u,0);
}
bool bfs()
{
    queue<int>q;
    memset(le,0,sizeof(le));
    le[s]=1;
    q.push(s);
    while(!q.empty())
    {
        int u=q.front();
        q.pop();
        for(int i=h[u];i;i=e[i].ne)
            if(e[i].va>0&&!le[e[i].to])
            {
                le[e[i].to]=le[u]+1;
                q.push(e[i].to);
            }
    }
    return le[t];
}
int dfs(int u,int f)
{
    if(u==t||!f)
        return f;

    int us=0;

```

```

for(int i=h[u];i&&us<f;i=e[i].ne)
    if(e[i].va>0&&le[e[i].to]==le[u]+1)
    {
        int t=dfs(e[i].to,min(e[i].va,f-us));
        e[i].va-=t;
        e[i^1].va+=t;
        us+=t;
    }
if(!us)
    le[u]=0;
return us;
}
int dinic()
{
    int re=0;
    while(bfs())
        re+=dfs(s,inf);
    return re;
}
int main()
{
    freopen("guard.in","r",stdin);
    freopen("guard.out","w",stdout);
    scanf("%d",&n);
    m=1;
    while(~scanf("%d%d%d",&a[m].a,&a[m].b,&a[m].c))
    {
        c[a[m].c]=max(c[a[m].c],a[m].a);//cout<<a[m].a<<" "<<a[m].b<<" "<<a[m].c<<endl;
        c[a[m].c+1>n?1:a[m].c+1]=max(c[a[m].c+1>n?1:a[m].c+1],a[m].b);
        m++;
    }
    m--;//cout<<m<<endl;
    s=0,t=101;
    for(int i=1;i<=n;i++)
        sum[i]=sum[i-1]+c[i];
    ans=sum[n];
    if((n&1)==0)
    {
        for(int i=1;i<=n;i++)
            for(int j=1;j<=c[i];j++)
            {
                if(i&1)
                    ins(s,sum[i-1]+j,1);
                else
                    ins(sum[i-1]+j,t,1);
            }
        for(int i=1;i<=m;i++)
        {
            if(a[i].c&1)
                ins(sum[a[i].c-1]+a[i].a,(a[i].c==n?0:sum[a[i].c])+a[i].b,inf);
            else
                ins((a[i].c==n?0:sum[a[i].c])+a[i].b,sum[a[i].c-1]+a[i].a,inf);
        }
    }
}

```

```

printf("%d\n",ans-dinic());
return 0;
}
int w,mn=1e9;
for(int i=1;i<=n;i++)
    if(c[i]<mn)
        {cout<<c[i]<<"? ? "<<endl;
         mn=c[i];
         w=i;
        }
for(int i=0;i<(1<mn);i++)
{
    memset(del,0,sizeof(del));
    memset(h,0,sizeof(h));
    cnt=1;
    int x=i,su=ans;
    for(int j=1;j<=c[w];j++)
    {
        if(x&1)
        {
            del[sum[w-1]+j]=1;
            for(int i=1;i<=m;i++)
                if(sum[a[i].c-1]+a[i].a==sum[w-1]+j||(a[i].c==n?
0:sum[a[i].c])+a[i].b==sum[w-1]+j)
                {
                    if(!del[sum[a[i].c-1]+a[i].a])
                        del[sum[a[i].c-1]+a[i].a]=1,su--;
                    if(!del[(a[i].c==n?0:sum[a[i].c])+a[i].b])
                        del[(a[i].c==n?0:sum[a[i].c])+a[i].b]=1,su--;
                }
            }
        else
            su--;
        x>>=1;
    }
    for(int i=1;i<=n;i++)
        if(i!=w)
            for(int j=1;j<=c[i];j++)
            {
                int now=i;
                if(i>w)
                    now--;
                if(now&1)
                    ins(s,sum[i-1]+j,1);
                else
                    ins(sum[i-1]+j,t,1);
            }
        for(int i=1;i<=m;i++)
            if(a[i].c!=w&&a[i].c!=(w-1==0?n:w-1)&&!del[sum[a[i].c-1]+a[i].a]&&!del[(a[i].c==n?
0:sum[a[i].c])+a[i].b])
                {cout<<"???"<<endl;
                 if((a[i].c-(a[i].c>w))&1)

                    ins(sum[a[i].c-1]+a[i].a,(a[i].c==n?0:sum[a[i].c])+a[i].b,inf);

```

```

        else
            ins((a[i].c==n?0:sum[a[i].c])+a[i].b,sum[a[i].c-1]+a[i].a,inf);
        }
        an=max(an,su-dinic());
    }
    printf("%d\n",an);
    return 0;
}

```

t2 Game

在你完成了上一个艰巨的任务之后，作为奖励，小A决定 ~~给你100分~~ 带你去打游戏。作为一个地地道道的土豪，小A玩的游戏当然很独特，他喜欢拼东西玩，但是拼接的元件都是特殊制造的，每个元件上有两个拼接口，不妨叫做 **A**口和 **B**口。由于设计的特殊性，任意两个元件，其中一个的 **A**口必定可以和另一个的 **B**口连接起来，并且其他拼接方式一定不能拼接住。

一盒玩具里会有 n 个元件，标记为 $1 \sim n$ ，并且有一个 $n \times n$ 的巨大表格告诉你任意两个元件之间的拼接关系。小A的目标就是把这些元件全部拼接到一起，如果可以的话最好拼接成一个环，如果不行的话一条链也可以接受。

然而由于元件太多小A已经玩不下去了，作为优秀员工的你当然要不露声色地帮小A完成这个游戏。

输入格式

第一行一个整数 n 表示原件个数。

接下来一个 $n \times n$ 的0/1表格，第 i 行第 j 列如果是1表示 i 的 **A**口可以和 j 的 **B**口连接起来，否则表示 j 的 **A**口可以和 i 的 **B**口连接起来。（主对角线上全部是0）

输出格式

第一行一个整数 t ，如果可以拼接成环则输出0，否则如果可以拼接成链输出1，如果都不行输出-1。

如果 $t \neq -1$ 接下来一行 n 个数表示一个合法的拼接序列，要保证前一个的 **A**口可以和后一个的 **B**口连接起来。

样例输入

```

5
0 0 1 1 1
1 0 1 1 0
0 0 0 1 0
0 0 0 0 1
0 1 1 0 0

```

样例输出

```

0
1 3 4 5 2

```

数据规模与约定

对于15%的数据， $n \leq 10$ 。

对于30%的数据， $n \leq 30$ 。

对于存在环的数据，如果你输出了一条链可以得到**40%**的分数。

对于**100%**的数据， $2 \leq n \leq 200$ 。

题解&反思

“假装它是个环”式暴力，38分。

思路比较迷，这个竞赛图，保证一定有一条贯穿全图的链**证明略**，过程如下：

tarjan缩点，如果只有一个联通块，那么一定有环。dfs一遍把环压到栈里输出即可。

否则就是链，按照tarjan出的belong分组dfs，同样压到栈里输出。

~~感冒子脑子不太清醒所以我在胡说些什么...~~

```

#include<iostream>
#include<cstdio>
#include<cstring>
using namespace std;
const int N=1e5;
int n,a[205][205],h[N],cnt,bl[N],dfn[N],low[N],tot,s[N],si[N],con,top,ans,c[N],l[N],r,zu,wo;
bool v[N];
struct qwe
{
    int ne,no,to;
}e[N<<1];
int read()
{
    int r=0;
    char p=getchar();
    while(p>'9' || p<'0')
        p=getchar();
    while(p>='0'&&p<='9')
    {
        r=r*10+p-48;
        p=getchar();
    }
    return r;
}
void add(int u,int v)
{
    cnt++;
    e[cnt].ne=h[u];
    e[cnt].no=u;
    e[cnt].to=v;
    h[u]=cnt;
}
void tarjan(int u)
{
    dfn[u]=low[u]=++tot;
    v[s[++top]=u]=1;
    for(int i=h[u];i;i=e[i].ne)
    {
        if(!dfn[e[i].to])
        {
            tarjan(e[i].to);
            low[u]=min(low[u],low[e[i].to]);
        }
        else if(v[e[i].to])
            low[u]=min(low[u],dfn[e[i].to]);
    }
    if(dfn[u]==low[u])
    {
        while(s[top]!=u)
        {
            bl[s[top]]=u;
            si[u]++;
            v[s[top--]]=0;
        }
    }
}

```

```

    }
    bl[s[top]]=u;
    si[u]++;
    v[s[top--]]=0;
}
}
void wk(int u,int la,int num)
{
    s[++top]=u;
    v[u]=1;
    num--;
    if(!num&&(u==la||a[u][la]))
    {
        wo=1;
        return;
    }//cout<<"OK"<<endl;
    for(int i=h[u];i=e[i].ne)
        if(!v[e[i].to]&&zu==bl[e[i].to])
        {
            wk(e[i].to,la,num);
            if(wo)
                return;
        }
    top--;
    v[u]=0;
}
int main()
{
    n=read();//cout<<"???"<<endl;
    for(int i=1;i<=n;i++)
        for(int j=1;j<=n;j++)
        {
            a[i][j]=read();
            if(a[i][j])
                add(i,j);
        }
    for(int i=1;i<=n;i++)
        if(!dfn[i])
            tarjan(i);
    for(int i=1;i<=n;i++)
        if(bl[i]!=bl[1])
            ans=1;
    printf("%d\n",ans);
    for(int i=1;i<=cnt;i++)
        if(bl[e[i].no]!=bl[e[i].to])
            c[bl[e[i].to]]++;
    for(int i=1;i<=n;i++)
        if((bl[i]==i)&&(!c[i]))
            l[++r]=i;
    for(int i=1;i<=r;i++)
        for(int j=1;j<=n;j++)
            if(j==bl[l[i]])

                for(int k=h[j];k;k=e[k].ne)

```



```

        if(bl[e[k].no]!=bl[e[k].to])
        {
            c[bl[e[k].to]]--;
            if(!c[bl[e[k].to]])
                l[++r]=e[k].to;
        }//cout<<"OK"<<endl;
    for(int i=1;i<=r;i++)
    {
        //cout<<top<<endl;
        zu=l[i];
        wo=0;
        wk(l[i],l[i],si[l[i]]);
    }
    for(int i=1;i<=n;i++)
        printf("%d ",s[i]);
    return 0;
}

```

t3 entertainment

题目描述

在连续帮小A完成了各种任务之后，你有幸得到了 **200分** 和小A一起度过 **n** 天假期的机会。你想要在这 **n** 天里让小A尽量的高兴，说到乐趣那当然是吃和睡了，为了充分享受，你决定让小A一整天要么吃饭要么睡觉。于是你根据各种参数外加玄学计算出了小A在将来 **n** 天里，每一天吃饭或者睡觉的快乐值。然而我们不能只吃饭或者只睡觉，所以在每个连续的 **k** 天里，至少要有 **ms** 天在睡觉， **me** 天吃饭。你希望在满足条件的情况下，最大化小A的快乐值之和，（这样下次你就可以自己过假期了）。

输入格式

第一行四个整数 **n, k, ms, me** 。

第二行 **n** 个整数，表示未来每一天睡觉可以得到的快乐值。

第三行 **n** 个整数，表示未来每一天吃饭可以得到的快乐值。

输出格式

一行一个整数表示答案。

样例输入

```

10 4 1 2
1 2 3 4 5 6 7 8 9 10
10 9 8 7 6 5 4 3 2 1

```

样例输出

```

69

```

数据规模与约定

对于 **30%** 的数据， **$n \leq 10$** 。

对于100%的数据， $1 \leq n \leq 1000$; $1 \leq me, ms \leq k \leq n$ 。

题解&反思

爆搜理论30，maye暴力分还要开long long。-20

所以所有数的范围都很重要啊。

程序里把 $n-k+1$ 当成 s ， $n-k+2$ 当成 t 即可，单独拎出来的 st 只是为了限流。

首先全部钦定为睡觉，然后把吃饭改成“吃饭和睡觉的差”，然后像差分约束一样列不等式，根据不等式建立最大费用最大流模型。

我到底在说什么啊

```

#include<iostream>
#include<cstdio>
#include<queue>
#include<cstring>
using namespace std;
const int N=10005;
const long long inf=1e18;
int n,k,ms,me,h[N],q[N],fr[N],cnt=1,s,t;
long long ans,a[N],b[N],dis[N];
bool v[N];
struct qwe
{
    int ne,no,to,v;
    long long c;
}e[N<<1];
void add(int u,int v,int w,long long c)
{
    cnt++;
    e[cnt].ne=h[u];
    e[cnt].no=u;
    e[cnt].to=v;
    e[cnt].c=c;
    e[cnt].v=w;
    h[u]=cnt;
}
void ins(int u,int v,int w,long long c)
{
    add(u,v,w,c);
    add(v,u,0,-c);
}
bool spfa()
{
    for(int i=s;i<=t;i++)
        dis[i]=-inf;
    queue<int>q;
    q.push(s);
    dis[s]=0;
    v[s]=1;
    while(!q.empty())
    {
        int u=q.front();
        q.pop();
        v[u]=0;
        for(int i=h[u];i;i=e[i].ne)
            if(e[i].v&&dis[e[i].to]<dis[u]+e[i].c)
            {
                fr[e[i].to]=i;
                dis[e[i].to]=dis[u]+e[i].c;
                if(!v[e[i].to])
                {
                    v[e[i].to]=1;
                    q.push(e[i].to);
                }
            }
    }
}

```

```

    }
}
return dis[t] != -inf;
}
void mcf()
{
    int x=1e9;
    for(int i=fr[t];i;i=fr[e[i].no])
        x=min(x,e[i].v);
    for(int i=fr[t];i;i=fr[e[i].no])
    {
        e[i].v-=x;
        e[i^1].v+=x;
        ans+=x*e[i].c;
    }
}
int main()
{
    freopen("entertainment.in","r",stdin);
    freopen("entertainment.ans","w",stdout);
    scanf("%d%d%d%d",&n,&k,&ms,&me);
    s=0;t=n+1;
    int tmp=k-ms-me;
    for(int i=1;i<=n;i++)
    {
        scanf("%lld",&a[i]);
        ans+=a[i];
    }
    for(int i=1;i<=n;i++)
    {
        scanf("%lld",&b[i]);
        b[i]=b[i]-a[i];
    }
    for(int i=1;i<=min(k,n-k);i++)
        ins(n-k+1,i,1,b[i]);
    for(int i=n-k+1;i<=k;i++)
        ins(n-k+1,n-k+2,1,b[i]);
    ins(n-k+1,1,tmp,0);
    ins(s,n-k+1,k-ms,0);
    ins(n-k+2,t,k-ms,0);
    ins(n-k,n-k+2,tmp,0);
    for(int i=max(k+1,n-k+1);i<=n;i++)
        ins(i-k,n-k+2,1,b[i]);
    for(int i=k+1;i<=n-k;i++)
        ins(i-k,i,1,b[i]);
    for(int i=2;i<=n-k;i++)
        ins(i-1,i,tmp,0);
    while(spfa())
        mcf();
    printf("%lld\n",ans);
    return 0;
}

```

总结

$100+38+10=148$

t3有20分不该丢，t2仔细想想应该能写正解。

距目标还有152分。