

B - The Accomodation of Students(二分图判断&匹配)

题意：给定若干条有向边，判断该图是否为二分图，并且求最大二分匹配。

思路：用染色法判断二分图+匈牙利算法解决。 时间复杂度： $O(n^3 + n^2)$

AC代码:

```
#include<cstdio>
#include<cstring>
#include<queue>
using namespace std;
const int N=2e2+10;
bool g[N][N];
int n,m,vis[N],mh[N],col[N];
void init(int n){ //初始化
    for(int i=1;i<=n;i++)
        vis[i]=mh[i]=col[i]=0;
    memset(g,0,sizeof g);
}
int bfs(){ //染色法判断二分图
    queue<int>q;
    q.push(1);
    col[1]=1;
    while(q.size()){
        int u=q.front();q.pop();
        for(int v=1;v<=n;v++)
            if(g[u][v]){
                if(!col[v]) col[v]=-col[u],q.push(v); //放到对立集.
                else if(col[u]==col[v]) return 0;
            }
    }
    return 1;
}
int find(int u){ //找妹子.
    for(int i=1;i<=n;i++){
```

```

        if(g[u][i]&&!vis[i]){
            vis[i]=1;
            if(!mh[i]||find(mh[i]))
            {
                mh[i]=u;
                return 1;
            }
        }
    }
    return 0;
}

int fun(){ //计算答案.
    int ans=0;
    for(int i=1;i<=n;i++)
    {
        memset(vis,0,sizeof vis);
        ans+=find(i);
    }
    return ans;
}

int main(){
    while(~scanf("%d%d",&n,&m)){
        init(n);
        for(int i=1,u,v;i<=m;i++){
            scanf("%d%d",&u,&v);
            g[u][v]=1;
        }
        if(!bfs()) puts("No");
        else printf("%d\n",fun());
    }
    return 0;
}

```

二分图性质：不含奇环。

证明：反证法。

假设存在一个奇环： $v_1, v_2, v_3 \dots v_{2k-1}, k \in N^+$

任意相邻两点有边连接，且 v_1, v_{2k-1} 有一条边相邻。

假设 v_1 属于 V_x 集合, 依次类推 $v_2 \in V_y, v_3 \in V_x \dots$

可以知道编号为奇数的结点都属于 V_x , 编号为偶数的结点都属于 V_y .

因为 v_1, v_{2k-1} 相连, 且 v_1, v_{2k-1} 都属于 V_x , 与二分图相连结点属于不同点集的定义矛盾, 所以即证二分图不含奇环。

失忆药水(二分图结论)

[传送门](#)

挺好的一个图论结论题,

n 阶无向完全图, 求最少删去多少边得到二分图。

也就是最多有多少边能组成二分图。

显然两边取一半顶点是最优的情况。

n 是偶数是取两个 $\frac{n}{2}$, 当 n 是奇数时取 $\frac{n}{2}$ 和 $\frac{(n-1)}{2}$

所以答案是
$$\begin{cases} \frac{n(n-1)}{2} - \frac{n}{2} \times \frac{n}{2}, n \text{ 为偶数} \\ \frac{n(n-1)}{2} - \frac{(n-1)}{2} \times \frac{n}{2}, n \text{ 为奇数} \end{cases}$$