题意：在一个有向图中，两个人轮流搬运芯片，芯片初始位于点s，谁最后五路可走谁输，但是在这场游戏中，后手在每次轮到他的时候他正好都在睡觉，于是先手都会偷偷帮他走，然后问你先手在最优状态下的结局，如果先手赢，要输出先手总共走的路径。

题解：其实翻译过来就是问你一个有向图中是否存在一条走奇数边能走到一个出度为0的点的路径，如果存在就输出路径，当然，假如不存在并且从s点走存在环，则一定就是平局，否则就是lose，因为题上说最多走1000000次，我们可以先广搜判断出环的情况，这样能先保证最坏也是平局，然后我们dfs爆搜，找先手能赢的路径，但是此时会出现问题，因为我有可能先走一个环，再往下走正好能走奇数个边（这里可以证明如果走环这个环一定只走一次），这也是我狂wa28的原因，因此我们可以开一个2维的标记数组，来标记当前点之前走了奇数条边或者走了偶数条边到达这里的，这样就能很好解决了，剩下的就是用last标记一下走过的路径即可。

#include <bits/stdc++.h>

using namespace std;

typedef long long ll;

const int maxn=1000010;

vector<int> q[maxn];

queue<int> q1;

short mark;

int sum;

int path[maxn];

bool vis[maxn][2];

void dfs(int u,int num)//num代表从起点到当前点走了几条边

{

if(mark==1)

return ;

path[num]=u;

//path标记走过的路径

vis[u][num&1]=1;//表示能通过走奇数/偶数 条边到达u

int v;

for(int i=0;i<q[u].size();i++)

{

v=q[u][i];

if(vis[v][(num+1)&1])continue;//这里注意因为这里是看以前是否走过

//比如我现在num是奇数,那么我从u到v，num就变成偶数了

//所以我要看偶数步到v是否走过，而不是看当前的奇数num

dfs(v,num+1);

}

if(num&1 && q[u].size()==0) //走到头了

//num是奇数就是走了奇数条边

{

sum=num;

mark=1;

return ;

}

if(mark==0)

{

mark=2;

return;

}

}

int main()

{

// freopen("input.txt","r",stdin);

int n,m,s,t,x;

cin>>n>>m;

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

{

cin>>t;

for(int j=1;j<=t;j++)

{

cin>>x;

q[i].push\_back(x);

}

}

cin>>s;

q1.push(s);

int now;

int cnt(0);

mark=0;

while(q1.empty()==0)//判断是否有环

{

now=q1.front();

q1.pop();

for(int i=0;i<q[now].size();i++)

{

cnt++;

q1.push(q[now][i]);

}

if(cnt>=1000000)

{

mark=3;

break;

}

}

dfs(s,0);

if(mark==1)

{

puts("Win");

for(int i=0;i<sum;i++)

cout<<path[i]<<' ';

cout<<path[sum]<<endl;

return 0;

}

if(mark==3)//mark=3说明有平局

{

puts("Draw");

return 0;

}

if(mark==2)

puts("Lose");

return 0;

}