

J Triangle War

题意：两个人轮流在由九个小正三角形组成的一个大正三角形上选取边。如果当前选取的边是某个小三角形的最后一条被选取的边，那么这个小三角形就归当前人所有，并且此人将再进行一次选取(即跳过对手接下来的一回合)。所以边都被选取后，占有的小三角形多的为胜者。现在给出已经进行的 m 次选择，问两人都采取最优策略的情况下谁是胜者。

思路：首先注意到，供选取的边只有 18 条，也就是说只会有 2^{18} 种状态。

那么对每条边按状态编号，对 9 个三角形则分别记录其由哪三条边组成，这样一来边和三角形就都成为了 2^{18} 个状态的一部分。

假如当前状态为 $state$ ， x 为一条边，那么判断 $state \& x$ 即可知道当前状态是否包含 x 这条边， $state \mid x$ 就代表在当前状态加入 x 这条边之后的状态。

我们可以利用极大极小博弈的思想，先手获得的三角形数与后手获得的三角形数的差值就是两人的收益函数，那么先手总想使函数值更大，后手相反。

那么我们只要根据题目的输入，按规则模拟，然后对状态记忆化搜索即可，最后两人获得的三角形数目差值的正负就是两人的胜负。

Memory: 1708 KB

Time: 313 MS

Language: G++

Result: Accepted

```
#include <stdio>
#include <string>
#include <algorithm>

using namespace std;

const int MIN = -100000000;
const int MAX_S = 0x40000;

const int pow2[18]={0x1,0x2,0x4,0x8,0x10,0x20,0x40,0x80,
0x100,0x200,0x400,0x800,0x1000,0x2000,0x4000,0x8000,0x10000,0x20000};

const int line[18][2]={
    {1,2},{1,3},{2,3},
    {2,4},{2,5},{3,5},
    {3,6},{4,5},{5,6},
    {4,7},{4,8},{5,8},
    {5,9},{6,9},{6,10},
    {7,8},{8,9},{9,10}
};

const int tri[9][3]={
    {pow2[0],pow2[1],pow2[2]},
    {pow2[3],pow2[4],pow2[7]},
    {pow2[2],pow2[4],pow2[5]},
```

```

        {pow2[5],pow2[6],pow2[8]},
        {pow2[9],pow2[10],pow2[15]},
        {pow2[7],pow2[10],pow2[11]},
        {pow2[11],pow2[12],pow2[16]},
        {pow2[8],pow2[12],pow2[13]},
        {pow2[13],pow2[14],pow2[17]},
    };

int f[MAX_S+5];

int find(int state,int pos)
{
    int flag,num=0;
    for(int i=0;i<9;i++)
    {
        flag=0;
        int id = -1;
        for(int j=0;j<3;j++)
        {
            if(pow2[pos] & tri[i][j])    id = j;
            else if(state & tri[i][j])    flag++;
        }
        if(~id && flag == 2)    num++;
    }
    return num;
}

int gao(int state)
{
    if(f[state]!=MIN)    return f[state];
    int mmax = MIN;
    for(int i=0;i<18;i++)
    {
        if(!(state&pow2[i]))
        {
            int tmp = find(state,i);
            tmp += (tmp>0?-1)* gao(state | pow2[i]);
            mmax = max(mmax,tmp);
        }
    }
    return f[state] = mmax;
}

int main(void)
{

```

```

int t,n,tmp;
scanf("%d",&t);
fill(f,f+MAX_S,MIN);
f[MAX_S-1]=0;
for(int cas=1;cas<=t;cas++)
{
    scanf("%d",&n);
    int player = 1,state = 0,cnt = 0;
    for(int i=0;i<n;i++)
    {
        int x,y;
        scanf("%d%d",&x,&y);
        for(int j=0;j<18;j++)
            if(x == line[j][0] && y==line[j][1])
            {
                tmp = find(state,j);
                state |= pow2[j];
                break;
            }
        if(tmp) cnt += player * tmp;
        else player *= -1;
    }
    printf("Game %d: %c wins.\n",cas,((cnt+player*gao(state))>0?'A':'B'));
}
return 0;
}

```