

**题目链接：**[Galaxy at War](#)

**题意：**在一个给定的 $n * m$ 的棋盘内的若干个位置 $(x_i, y_i)$ 有 $w_i$ 个水晶，同时有 $t$ 个冥想球和 $s$ 个污染源，每一次可以选定一个位置 $(u, v)$ ，将 $t$ 个水晶移动到 $(u + 1, v)$ 或者 $(u, v + 1)$ ，如果这个位置有冥想球，那么可以在 $(u + 1, v)$ 和 $(u, v + 1)$ 处制造 $t$ 个水晶。如果这个位置有污染源，那么只有 $\lfloor \frac{t}{2} \rfloor$ 个水晶能保存下来。 $(n, m)$ 处有一个污染源， $(n - 1, m)$ 处有一个冥想球。并且每一个冥想球和每一个污染源之间的距离是奇数。问维德和尤达大师轮流行动，不能行动的那一方输掉游戏，问维德是否能取得胜利。

**题解：**这个游戏是一个简单的阶梯博弈，因为在 $(n, m)$ 处有污染源且所有冥想球与污染源的距离为奇数，那么我们只需要考虑与冥想球曼哈顿距离为偶数的点即可。我们把这些点的水晶个数异或起来，如果是0，代表先手按照Nim博弈的取法取就行，因为后手不论怎么采取策略，先手一定会有办法移动水晶使得局面必败（证明）。所以我们对于每一堆水晶，判断其是否和某一个它能够移动到的冥想球的曼哈顿距离为偶数，然后把这堆的数目异或得到答案。由于从题意可以推出每一对冥想球之间的距离一定是偶数，所以我们直接判断是否和 $(n - 1, m)$ 处的冥想球距离为偶数即可，对于 $x = n$ 的水晶记录 $x = n$ 的冥想球 $y$ 坐标最大值特判即可。

```
1  #include <iostream>
2  #include <stdio.h>
3  #include <vector>
4  #include <algorithm>
5
6  using namespace std;
7
8  const int N = 500005;
9  pair<int, int> med[N], pol[N];
10 struct Ct{
11     int x, y, w;
12     Ct() {}
13     Ct(int x, int y, int w) : x(x), y(y), w(w) {}
14 };
15 vector<Ct> c;
16
17 int main() {
18     int T, n, m, k, x, y, w, t, s, mx;
19     scanf("%d", &T);
20     while(T--) {
21         mx = 0;
22         scanf("%d %d %d", &n, &m, &k);
23         c.clear();
24         for(int i = 1; i <= k; i++) {
25             scanf("%d %d %d", &x, &y, &w);
26             c.push_back(Ct(x, y, w));
27         }
```

```
27     }
28     scanf("%d",&t);
29     for(int i=1;i<=t;i++){
30         scanf("%d %d",&med[i].first,&med[i].second);
31         if(med[i].first==n)
32             mx=max(mx,med[i].second);
33     }
34     scanf("%d",&s);
35     for(int i=1;i<=s;i++){
36         scanf("%d %d",&pol[i].first,&pol[i].second);
37     }
38     int ans=0;
39     for(int i=0;i<k;i++){
40         if(c[i].x<n&&(abs(c[i].x-n+1)+abs(c[i].y-m))%2==0)
41             ans^=c[i].w;
42         else if(c[i].x==n&&c[i].y<=mx&&abs(c[i].y-mx)%2==0)
43             ans^=c[i].w;
44     }
45     puts(ans?"win":"lose");
46 }
47 return 0;
48 }
```

