题意：编号1-n的盒子，当你的编号满足（是编号!编号!编号！） B<A && (A+B)%2=1 && (A+B)%3=0的时候你可以从a中取至少一个石头到b不能取的人输问你输赢。

思路：   
显然 1 3 4 这三个位置无法移到其他位置。每个位置移到下个位置后移动步数的奇偶性改变，这就符合上面所说的阶梯博弈了。   
打表容易看出得出模6为0、2、5的位置移动步数为奇，其余为偶数。   
根据上面我们可以得出移动位置是偶数步的位置上的牌数是无关紧要的：因为不论对方将偶数步位置上的牌作如何操作，我都可以把他移动的牌再往后移动一步。（也就是说偶数步上的位置的牌我可以保证两个人对其的操作数是偶数次，从而不影响到奇数步位置上的牌的状态。）

奇数步位置上的牌可以直接移到最终位置：

①、如果直接把某一奇数步位置上的某些牌移到1、3、4这些终点位置，就相当于取石子游戏中从一堆石子里取走一些石子；   
②、如果把一些牌移到偶数步位置，之前说了偶数步位置上的牌可以无视的，我们也可以把它看作是从这堆石子里取走了一些石子。   
所以就是把奇数步位置上的异或起来就好了。   
打表：

打表代码：

#include <bits/stdc++.h>

typedef long long ll;

using namespace std;

int main()

{

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

int a[100];

a[1] = 0; a[3] = 0; a[4] = 0;

for(int i = 2; i < 100; i++)

{

for(int j = 1; j < i; j++)//i是A，j是B

{

if((i+j)%2 == 1&&(i+j)%3 == 0)

{

a[i] = a[j] + 1;//注意是a[i]=a[j]+1而不是a[j]=a[i]+1,因为是从A中取东西放到B，也就是A比B到达终态多了一步

break;

}

}

}

for(int i = 1; i < 100; i++)

if(a[i]%2 != 0)//奇数

//printf("a[%d] = %d\n",i%6,a[i]);

cout<<i%6<<"\n";

return 0;

}

题目代码

#include<stdio.h>

#include<string.h>

#include<iostream>

using namespace std;

int main()

{

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

int T,n,x;

cin>>T;

for(int o=1;o<=T;o++)

{

cin>>n;

int ans(0);

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

{

cin>>x;

if(i%6==0 ||i%6==2||i%6==5)//注意是i而不是x

ans^=x;

}

printf("Case %d: ",o);

if(ans==0)

puts("Bob");

else puts("Alice");

}

return 0;

}