题意:Alice和Bob轮流取N堆石子，每堆S[i]个，Alice先，每一次可以从任意一堆中拿走任意个石子，也可以将一堆石子分为两个小堆。先拿完者获胜。

数据范围: (1 ≤ N ≤ 10^6, 1 ≤ S[i] ≤ 2^31 - 1).

思路: 此题为博弈中的-----取走-分割游戏（这种游戏允许取走某些东西，然后将原来的一个游戏分成若干个相同的游戏）

由于数据范围,不能直接求sg值只能打表找规律;

Lasker's Nim游戏：每一轮允许两会中操作之一：①、从一堆石子中取走任意多个，②、将一堆数量不少于2的石子分成都不为空的两堆。

分析：很明显：sg(0) = 0，sg(1) = 1。

状态2的后继有：0，1和（1，1），他们的SG值分别为0，1，0，所以sg(2) =2。

状态3的后继有：0、1、2、（1，2），他们的SG值分别为0、1、2、3，所以sg(3) = 4。

状态4的后继有：0、1、2、3、（1，3）和（2，2），他们的SG值分别为0，1，2，4，5，0，所以sg(4) = 3.

打sg函数表代码

#include<stdio.h>

#include<string.h>

int sg[10000];

int main()

{

sg[0]=0;

sg[1]=1;

for(int i=2;i<=1000;i++)

{

bool vis[1010]={false};

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

{

vis[sg[j]]=true;//取石子

if(j!=0&&j!=i)

vis[sg[j]^sg[i-j]]=true;//拆分

}

int j=0;

while(vis[j]!=0)

j++;

sg[i]=j;

}

for(int i=1;i<=1000;i++)//注意是从1开始而不是0

{

printf("%d ",sg[i]);

//if(i%4==0)printf("\n");

}

}

代码：

注意这样每读入一个x，用函数求其值返回就行了，不用再用个a数组存一下，或者先读进来再存

#include <bits/stdc++.h>

typedef long long ll;

using namespace std;

int find\_sg(int x)

{

if(x%4==0)

return x-1;

else if(x%4==3)return x+1;

else return x;

}

int main()

{

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

int T,n,x;

cin>>T;

while(T--)

{

cin>>n;

int ans=0;

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

{

cin>>x;

ans^=find\_sg(x);

}

if(ans==0)

puts("Bob");

else puts("Alice");

}

return 0;

}