**高僧斗法**

问题描述

　　古时丧葬活动中经常请高僧做法事。仪式结束后，有时会有“高僧斗法”的趣味节目，以舒缓压抑的气氛。  
　　节目大略步骤为：先用粮食（一般是稻米）在地上“画”出若干级台阶（表示N级浮屠）。又有若干小和尚随机地“站”在某个台阶上。最高一级台阶必须站人，其它任意。(如图1所示)  
　　两位参加游戏的法师分别指挥某个小和尚向上走任意多级的台阶，但会被站在高级台阶上的小和尚阻挡，不能越过。两个小和尚也不能站在同一台阶，也不能向低级台阶移动。  
　　两法师轮流发出指令，最后所有小和尚必然会都挤在高段台阶，再也不能向上移动。轮到哪个法师指挥时无法继续移动，则游戏结束，该法师认输。  
　　对于已知的台阶数和小和尚的分布位置，请你计算先发指令的法师该如何决策才能保证胜出。

输入格式

　　输入数据为一行用空格分开的N个整数，表示小和尚的位置。台阶序号从1算起，所以最后一个小和尚的位置即是台阶的总数。（N<100, 台阶总数<1000）

输出格式

　　输出为一行用空格分开的两个整数: A B, 表示把A位置的小和尚移动到B位置。若有多个解，输出A值较小的解，若无解则输出-1。

样例输入

1 5 9

样例输出

1 4

样例输入

1 5 8 10

样例输出

1 3

锦囊1

博弈论，NIM取子游戏。

本题的C++参考代码如下：

//2014年1月12日12:29:37

#include <stdio.h>

#include <stdlib.h>

#define N 1050

int a[N]; //存放输入数据

int s[N]; //虚拟石子

int total = 0;

int flag= 0;

int i,j;

int main()

{

do

{

scanf("%d",&a[total++]);

if(total%2==0)

s[total/2] = a[total-1]-a[total-2]-1;

}while(getchar()!='\n');

for(i=1;i<=total/2;i++)

{

flag^=s[i];

}

//一开始就处于N态

if(!flag) printf("-1");

else

{

for(i=1;i<=total/2;i++)//从最小的开始遍历

{

int temp = s[i];

//左边上移减少数量

while(s[i]--)

{

flag = 0;

for(j=1;j<=total/2;j++) //再检测

flag^=s[j];

//把对手变成p态!

if(flag==0)

{

//第i个移动i步(移动到哪里)

//应该是从哪个位置移到哪个位置 跟哪个小和尚没关系

printf("%d %d\n",a[i\*2-2],a[i\*2-2]+temp-s[i]);

exit(0);

}

}

s[i] = temp;

//右边上移增加数量

while(s[i]-temp <a[i\*2]-a[2\*i-1]-1)

{

s[i]++;

flag = 0;

for(j=1;j<=total/2;j++) //再检测

flag^=s[j];

if(flag==0)

{

printf("%d %d\n",a[i\*2-1],a[i\*2-1]+s[i]-temp);

exit(0);

}

}

s[i]=temp;

}

}

return 0;

}

本题的C参考代码如下：

#include <stdio.h> //参考蓝桥杯贴吧 dezhonger

int main()

{

int a[105],b[105],i=0,j,k,count,sum;

char c;

while(1)

{

scanf("%d%c",&a[i++],&c);

if(c=='\n')

break;

}

count=i;

for(i = 0;i < count-1;i++)

b[i]=a[i+1]-a[i]-1;

b[count-1]=0;

sum=b[0];

for(i = 2;i < count;i = i+2)

sum^=b[i];

if(sum == 0)

printf("-1\n");

else

{

for(i = 0;i < count;i++)

for(j = 1;j <= b[i];j++)

{

b[i] -= j;

if(i!=0)

b[i-1]+=j;

sum = b[0];

for(k = 2;k < count;k = k+2)

sum ^= b[k];

if(sum == 0)

{

printf("%d %d\n",a[i],a[i]+j);

break;

}

b[i] += j;

if(i != 0)

b[i-1] -= j;

}

}

return 0;

}

/\*

1 3 5 7 12 14 17 26 38 45 66 100

66 84

\*/

本题的Java参考代码如下：

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

import java.util.StringTokenizer;

public class Main {

public static void main(String[] args) throws IOException {

BufferedReader bfr=new BufferedReader(new InputStreamReader(System.in));

StringTokenizer tok= new StringTokenizer(bfr.readLine());

int i=0,j,flag=1;

int[] monk=new int[108];

while(tok.hasMoreTokens()) monk[i++]=Integer.parseInt(tok.nextToken());

int N=i-1;

int[] A=new int[N];

for(i=0;i<N;i++) A[i]=monk[i+1]-monk[i]-1;

int sum=A[0];

for(i=2;i<N;i++,i++) sum^=A[i];

if(sum==0) System.out.println(-1);

else

{

for(i=0;i<N;i++)

{

for(j=1;j<=A[i];j++)

{

int s=sum;

A[i]-=j;

if(i>0) A[i-1]+=j;

if(i%2==0) { s^=(A[i]+j); s^=A[i]; }

else { s^=(A[i-1]-j); s^=A[i-1]; }

if(s==0) { System.out.println(monk[i]+" "+(monk[i]+j)); flag=0; break; }

else { A[i]+=j; if(i>0) A[i-1]-=j; }

}

if(flag==0) break;

}

}

}

}