一些敲简单的博弈论

一般的博弈论通过sg函数打表+找规律就可以解决 另外有一些不平等博弈/阶梯博弈需要一定的分析。

HDU1846-Brave Game

题目大意

给定一堆石子，有n个，两名玩家先后从石子堆中取石子儿，每次可以取走[1,m]颗，问先手是否有必胜策略

数据范围

$n\leq 1000,m\leq 1000$,多组数据≤ 100

题目分析

我们考虑求一个sg函数，打表后发现当且仅当n%(m+1)==0时先手没有必胜策略 在这里简单证明一下，这个游戏相当于我们小时候经常玩的取30，先手只需要保证把n%(m+1)的余数取走，然后不论后手取[1,m]中的几，先手都可以保证这一轮两人取的石子加起来等于m+1

代码

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <algorithm>
using namespace std;
const int N = 1005;
int sg[N];
bool vis[N];
int n,m;
void SG()
{
    memset(sg,0,sizeof(sg));
    for(int i = 0;i<=n;i++)
    {
        memset(vis,0,sizeof(vis));
        for(int j = max(i-m,0);j<i;j++)
            vis[sg[j]] = true;
        for(int j = 0;;j++)
            if(!vis[j])
                {sg[i] = j;break;}
    }
```

```
}
int main()
{
    int cas;
    scanf("%d",&cas);
    while(cas--)
    {
        scanf("%d%d",&n,&m);
        if(n%(m+1)==0)printf("second\n");
        else printf("first\n");
    }
    return 0;
}
```

HDU1847-Good Luck in CET-4 Everybody!

题目大意

给定一堆n个石子，每次可以从中取走2^k个石子，不能操作者负，问先手是否有必胜策略

数据范围

$n\leq 1000$

题目分析

这题连找规律都不用了…… 直接上sg函数就行了。

代码

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <algorithm>
using namespace std;
const int N = 1005;
int sg[N];
bool vis[N];
void SG()
{
    for(int i = 0;i<N;i++)
    {
```

```
        memset(vis,0,sizeof(vis));
        for(int j = 0;1<<j<=i;j++)
            vis[sg[i-(1<<j)]] = true;
        for(int j = 0;;j++)
            if(!vis[j])
                {sg[i] = j;break;}
    }
}
int main()
{
    int n;
    SG();
    while(scanf("%d",&n)!=EOF)
    {
        if(sg[n])printf("Kiki\n");
        else printf("Cici\n");
    }
    return 0;
}
```

HDU2516-取石子游戏

题目大意

给定一堆n个石子，两个人轮流取石子，每次取石子的个数在1~上一个人取的个数*2之间，并且规定第一轮不能把石子取光，问先手是否有必胜策略

数据范围

$n\leq 2^{31}$

题目分析

这个也算是比较经典的博弈问题。 我们分析一下这个博弈的情况，如果n=2□那么先手必败 如果n=3□如果先手取1，那么后手恰好可以取2，先手败，如果先手取2，那么后手可以取1，先手必败 如果n=4□这时候如果先手取1，那么后手只能取1或2，考虑不论剩下多少先手都可以全部取走，此时先手胜 然后考虑n=5的时候，如果先手取1，那么情况变成了n=4时的情况，如果先手取2，那么后手取走3就可以获胜。 以此类推，发现每次先手必败的状态都是他之前两个先手必败状态的石子个数加和。 所以我们只需要先预处理一个斐波那契数列，然后看n是否是这个数列中的某一项即可。 因为斐波那契数列增长是指数级的，所以最多50项，每次暴力找就行了

代码

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <algorithm>
using namespace std;
int f[100];
int main()
{
    f[0] = f[1] = 1;
    for(int i = 2;i<100;i++)
        f[i] = f[i-1]+f[i-2];
    int n;
    while(scanf("%d",&n)!=EOF)
    {
        if(!n)break;
        bool flag = false;
        for(int i = 0;i<100;i++)
            if(f[i]==n){flag = true;break;}
        if(flag)printf("Second win\n");
        else printf("First win\n");
    }
    return 0;
}
```

HDU2897-邂逅明下

题目大意

给出若干个游戏，游戏规则是现在有n个石子，每次可以取[p,q]间任意数目石子，如果小于等于p必须一次性全部取走，问是否有先手必胜策略

数据范围

数据组数$\leq 10^5$ $n,p,q\leq 65536$

题目分析

看到这么大的数据范围，我们必然是要求sg函数打表然后找规律了，每次把-[p,q]之间的sg值都标为vis,然后每次取最小的没出现过的自然数为当前局面的sg值，发现如果有n%(p+q)<=p的话那么其sg函数值一定是0，然后这个问题就可以完美解决啦

代码

```
#include <stdio.h>
int main()
{
    int n,p,q;
    while(scanf("%d%d%d",&n,&p,&q)!=EOF)
    {
        if(n%(p+q)>p||n%(p+q)==0)printf("WIN\n");
        else printf("LOST\n");
    }
}
```

HDU3032-Nim or not Nim?

题目大意

给定n堆石子，现在玩一个规则稍有不同的nim游戏，我们一次可以从一堆石子中取走若干个，也可以将一堆石子分为较小的两堆，问先手是否有必胜策略。

数据范围

$n\leq 10^6$,$s_i\leq 10^8$

题目分析

好，大范围继续sg函数打表找规律 sg函数这样设计，求值为i的sg函数时，把从0~i-1的局面的sg函数值标vis,然后把sg(1)^sg(i-1)sg(2)^sg(i-2).....等一系列局面的函数值标vis然后每次取当前没被标vis的最小自然数即可。 发现最后的sg(4k) = 4k-1,sg(4k+1) = 4k+1,sg(4k+2) = 4k+2,sg(4k+3) = 4k+4 然后是否有先手策略只需要把所有游戏的sg异或起来看是否为0即可

代码

```
#include <stdio.h>
int main()
{
    int cas;
    scanf("%d",&cas);
    while(cas--)
    {
        int n,x,ans = 0;
        scanf("%d",&n);
        while(n--)
        {
```

```
            scanf("%d",&x);
            if(x%4==0)x = x-1;
            else if(x%4==3)x = x+1;
            ans = ans^x;
        }
        if(ans)printf("Alice\n");
        else printf("Bob\n");
    }
}
```