**一.  巴什博奕（Bash Game）：**

**A和B一块报数，每人每次报最少1个，最多报4个，看谁先报到30。这应该是最古老的关于巴什博奕的游戏了吧。**

**其实如果知道原理，这游戏一点运气成分都没有，只和先手后手有关，比如第一次报数，A报k个数，那么B报5-k个数，那么B报数之后问题就变为，A和B一块报数，看谁先报到25了，进而变为20,15,10,5，当到5的时候，不管A怎么报数，最后一个数肯定是B报的，可以看出，作为后手的B在个游戏中是不会输的。**

**那么如果我们要报n个数，每次最少报一个，最多报m个，我们可以找到这么一个整数k和r，使n=k\*（m+1）+r，代入上面的例子我们就可以知道，如果r=0，那么先手必败；否则，先手必胜。**

**巴什博奕：只有一堆n个物品，两个人轮流从中取物，规定每次最少取一个，最多取m个，最后取光者为胜。**

**二.  威佐夫博弈（Wythoff Game）：**

**有两堆各若干的物品，两人轮流从其中一堆取至少一件物品，至多不限，或从两堆中同时取相同件物品，规定最后取完者胜利。**

**直接说结论了，若两堆物品的初始值为（x，y），且x<y，则另z=y-x；**

**记w=（int）[（（sqrt（5）+1）/2）\*z  ]；**

**若w=x，则先手必败，否则先手必胜。**

**三.  尼姆博弈（Nimm Game）：**

**尼姆博弈指的是这样一个博弈游戏：有任意堆物品，每堆物品的个数是任意的，双方轮流从中取物品，每一次只能从一堆物品中取部分或全部物品，最少取一件，取到最后一件物品的人获胜。**

**结论就是：把每堆物品数全部异或起来，如果得到的值为0，那么先手必败，否则先手必胜。**

**四.  斐波那契博弈：**

**有一堆物品，两人轮流取物品，先手最少取一个，至多无上限，但不能把物品取完，之后每次取的物品数不能超过上一次取的物品数的二倍且至少为一件，取走最后一件物品的人获胜。**

**结论是：先手胜当且仅当n不是斐波那契数（n为物品总数）**

**五．翻硬币游戏**

**一般的翻硬币游戏的规则是这样的：**

**N 枚硬币排成一排，有的正面朝上，有的反面朝上。我们从左开始对硬币按1 到N 编号。**

**第一，游戏者根据某些约束翻硬币，但他所翻动的硬币中，最右边那个硬币的必须是从正面翻到反面。例如，只能翻3个硬币的情况，那么第三个硬币必须是从正面翻到反面。如果局面是正正反，那就不能翻硬币了，因为第三个是反的。**

**第二，谁不能翻谁输。**

**有这样的结论：局面的SG 值为局面中每个正面朝上的棋子单一存在时的SG 值的异或和。即一个有k个硬币朝上，朝上硬币位置分布在的翻硬币游戏中，SG值是等于k个独立的开始时只有一个硬币朝上的翻硬币游戏的SG值异或和。比如THHTTH这个游戏中，2号、3号、6号位是朝上的，它等价于TH、TTH、TTTTTH三个游戏和，即sg[THHTTH]=sg[TH]^sg[TTH]^sg[TTTTTH].我们的重点就可以放在单个硬币朝上时的SG值的求法。**

**约束条件一：每次只能翻一个硬币。**

**一般规则中，所翻硬币的最右边必须是从正面翻到反面，因为这题是只能翻一个硬币，那么这个硬币就是最右边的硬币，所以，每次操作是挑选一个正面的硬币翻成背面。**

**对于任意一个正面的硬币，SG值为1。**

**有奇数个正面硬币，局面的SG值==1，先手必胜，有偶数个正面硬币，局面的SG值==0，先手必败。**

**约束条件二：每次能翻转一个或两个硬币。(不用连续)**

**每个硬币的SG值为它的编号，初始编号为0，与NIM游戏是一样的。**

**如果对于一个局面，把正面硬币的SG值异或起来不等于0，既a1^a2^a3^…^an==x,对于an来说一定有an'=an^x<an。**

**如果an'==0，意思就是说，把an这个值从式子中去掉就可以了。对应游戏，就是把编号为an的正面硬币翻成背面就可以了。因为an^x==0，而a1^a2^a3^…^an==x，即an^a1^a2^a3^…^an==0，即a1^a2^a3^…^an-1==0，只要在原来的x里面去掉an就可以了。**

**如果an'!=0，意思就是说，把an这个值从式子中去掉后再在式子中加上an'，an'<an。对应游戏，去掉an就是把编号为an的正面硬币翻成背面，加上an'，如果编号为an'的硬币是正面，我们就把它翻成背面，是背面就翻成正面，总之，就是翻转编号为an'的硬币。因为an^x!=0，所以an^a1^a2^a3^…^an!=0，即a1^a2^a3^…^an-1!=0，而这里的**

**an'=a1^a2^a3^…^an-1，所以在x中去掉an后，要对an'进行异或，也就是翻转，正转反，反转正。**

**约束条件三：每次必须连续翻转k个硬币。**

**我们以k==3为例。**

**我们计算的是个数为N的硬币中，其中最后一个硬币为正面朝上,的sg值。**

**当N==1时，硬币为：正，先手必输，所以sg[1]=0。**

**当N==2时，硬币为：反正，先手必输，所以sg[2]=0。**

**当N==3时，硬币为：反反正，先手必胜，所以sg[3]=1。**

**当N==4时，硬币为：反反反正，先手操作后为：反正正反，子状态局面的SG=0^1=1，那么sg[4]=0。**

**当N==5时，硬币为：反反反反正，先手操作后为：反反正正反，子状态局面的SG=1^0=1，那么sg[5]=0。**

**当N==6时，硬币为：反反反反反正，先手操作后为：反反反正正反，子状态局面的SG=0^0=0，那么sg[6]=1。**

**根据观察，可以知道，从编号为1开始，sg值为：001 001 001 001……**

**根据观察，可以知道，sg的形式为000…01 000…01，其中一小段0的个数为k-1。**

**约束条件4：每次翻动一个硬币后，必须翻动其左侧最近三个硬币中的一个，即翻动第x个硬币后，必须选择x-1，x-2，x-3中的其中一个硬币进行翻动，除非x是小于等于3的。（Subtraction Games）**

**当N==1时，硬币为：正，先手必赢，所以sg[1]=1。**

**当N==2时，硬币为：反正，先手必赢，因为先手可以翻成反反或正反，可能性为2，所以sg[2]==2。**

**当N==3时，硬币为：反反正，先手操作后可以为：反正**

**位置x：1  2  3  4  5  6  7  8  9  10  11  12  13  14...**

**sg[x]：  1  2  3  0  1  2  3  0  1   2     3     0    1     2…**

**这个与每次最多只能取3个石子的取石子游戏的SG分布一样，同样还有相似的这类游戏，约束条件5也是一样。**

**约束条件5：每次必须翻动两个硬币，而且这两个硬币的距离要在可行集S={1,2,3}中，硬币序号从0开始。(Twins游戏)**

**当N==1时，硬币为：正，先手必输，所以sg[0]=0。**

**当N==2时，硬币为：反正，先手必赢，所以sg[1]=1。**

**当N==3时，硬币为：反反正，先手必赢，所以sg[2]=2。**

**当N==4时，硬币为：反反反正，先手必赢，所以sg[3]=3。**

**当N==5时，硬币为：反反反反正，先手必输，所以sg[4]=0。**

**位置x：0  1  2  3  4  5  6  7  8  9  10  11  12  13  14...**

**sg[x]：  0  1  2  3  0  1  2  3  0  1   2     3     0    1     2…**

**约束条件6：每次可以翻动一个、二个或三个硬币。（Mock Turtles游戏）**

**初始编号从0开始。**

**当N==1时，硬币为：正，先手必胜，所以sg[0]=1.**

**当N==2时，硬币为：反正，先手必赢，先手操作后可能为：反反或正反，方案数为2，所以sg[1]=2。**

**当N==3时，硬币为：反反正，先手必赢，先手操作后可能为：反反反、反正反、正反正、正正反，方案数为4，所以sg[2]=4。**

**位置x：0  1  2  3  4   5    6   7    8     9  10  11  12  13  14...**

**sg[x]：  1  2  4  7  8  11 13 14  16  19  21  22  25  26  28…**

**看上去sg值为2x或者2x+1。我们称一个非负整数为odious，当且仅当该数的二进制形式的1出现的次数是奇数，否则称作evil。所以1，2，4，7是odious因为它们的二进制形式是1,10,100,111.而0,3,5,6是evil，因为它们的二进制形式是0,11,101,110。而上面那个表中，貌似sg值都是odious数。所以当2x为odious时，sg值是2x，当2x是evil时，sg值是2x+1.**

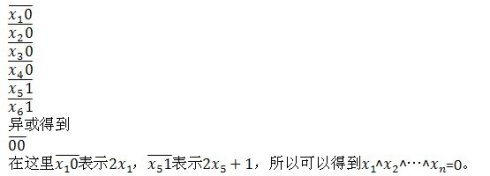
**这样怎么证明呢？我们会发现发现，**

**evil^evil=odious^odious=evil**

**evil^odious=odious^evil=odious**

**假设刚才的假说是成立的，我们想证明下一个sg值为下一个odious数。注意到我们总能够在第x位置翻转硬币到达sg为0的情况；通过翻转第x位置的硬币和两个其它硬币，我们可以移动到所有较小的evil数，因为每个非零的evil数都可以由两个odious数异或得到；但是我们不能移动到下一个odious数，因为任何两个odious数的异或都是evil数。**

**假设在一个Mock Turtles游戏中的首正硬币位置x1,x2,…,xn是个P局面，即sg[x1]^…^sg[xn]=0.那么无可置疑的是n必定是偶数，因为奇数个odious数的异或是odious数，不可能等于0。而由上面可知sg[x]是2x或者2x+1，sg[x]又是偶数个，那么x1^x2^…^xn=0。相反，如果x1^x2^…^xn=0且n是偶数，那么sg[x1]^…^sg[xn]=0。这个如果不太理解的话，我们可以先这么看下。2x在二进制当中相当于把x全部左移一位，然后补零，比如说2的二进制是10，那么4的二进制就是100。而2x+1在二进制当中相当于把x全部左移一位，然后补1，比如说2的二进制是10，5的二进制是101。现在看下sg[x1]^…^sg[xn]=0，因为sg[x]是2x或者2x+1，所以式子中的2x+1必须是偶数个（因为2x的最后一位都是0,2x+1的最后一位都是1，要最后异或为0,2x+1必须出现偶数次）。实际上的情况可能是这样的:**



**MT游戏当中的P局面是拥有偶数堆石子的Nim游戏的P局面。**

**约束条件7：每次可以连续翻动任意个硬币，至少翻一个。（Ruler游戏）**

**初始编号从1开始。**

**那么这个游戏的SG函数是g(n)=mex{0,g(n-1),g(n-1)^g(n-2),…,g(n-1)^…^g(1)}**

**根据SG函数可以得到SG值表如下。**

**位置x：1  2  3  4  5  6  7  8  9  10  11  12  13  14  15   16...**

**g(x):       1  2  1  4  1  2  1  8  1    2    1     4    1    2     1   16…**

**所以sg值为x的因数当中2的能达到的最大次幂。比如14=2\*7，最大1次幂，即2；16=2\*2\*2\*2，最大4次幂，即16。**

**这个游戏成为尺子游戏是因为SG函数很像尺子上的刻度。**

**约束条件8：每次必须翻转4个对称的硬币，最左与最右的硬币都必须是从正翻到反。（开始的时候两端都是正面）（Grunt游戏）**

**这是Grundy游戏的变种，初始编号从0开始。**

**当首正硬币位置为0,1,2时是terminal局面，即 终结局面，sg值都是0。当首正硬币位置n大于等于3的时候的局面可以通过翻0,x,n-x,n四个位置得到(其中x<n/2可保证胜利)。**

**这就像是把一堆石子分成两堆不同大小石子的游戏，也就是Grundy游戏。**

**Sg函数**

1. //f保存每次能取石子的情况
2. **int** sg[N],f[N],rec[N];
3. **void** getSg()
4. {
5. sg[0] = 0;
6. **for**( **int** i = 1 ; i < 1001 ; ++i ){
7. memset( rec , 0 , **sizeof**(rec) );
8. **for**( **int** j = 1 ; f[j] <= i ; ++j )
9. rec[sg[i-f[j]]] = 1;
10. **for**( **int** k = 0 ; k <= i ; ++k ){
11. **if**( rec[k] == 0 ){
12. sg[i] = k; **break**;
13. }
14. }
15. }
16. }