k倍动态减法

转载请注明出处，谢谢 http://blog.csdn.net/ACM\_cxlove?viewmode=contents           by---cxlove

神牛曹钦翔的论文：从“k倍动态减法游戏”出发探究一类组合游戏问题

反正偶是没看懂。。。。

两人取一堆n个石子 先手不能全部取完 之后每人取的个数不能超过另一个人上轮取的数\*K

给n，K判断先手必胜并求第一步

博弈题

这题的思考过程非常有意义。

当k=1的时候 可知必败局面都是2^i  将n分解成二进制，然后先手取掉最后一个1.然后对方必然无法去掉更高的1，而对方取完我方至少还能拿掉最后一个1 导致对方永远取不完。

当k=2的时候，必败局面都是斐波那契数列。利用“先手去掉最后一个1，则后手必不能去掉更高阶的1导致取不完”的思想，斐波那契数列有一个非常好的性质就是：任意一个整数可以写成斐波那契数列中的不相邻的项的和，于是将n写成这种形式，先取走最后一个1，对方能取的数是这个数\*2，小于高2位的1，所以取不完。

当K的时候， 想办法构造数列，将n写成数列中一些项的和，使得这些被取到的项的相邻两个倍数差距>k 那么每次去掉最后一个1 还是符合上面的条件。设这个数列已经被构造了i 项，第 i 项为a[ i ]，前 i 项可以完美对1..b[ i ] 编码使得每个编码的任意两项倍数>K 那么有

a[ i+1 ] = b[ i ] + 1;这是显然的 因为b[ i ] + 1没法构造出来，只能新建一项表示

然后计算b[ i+1] 既然要使用 a[ i+1 ] 那么下一项最多只能是某个 a[ t ] 使得 a[ t ] \* K < a[ i+1 ] 于是

b[ i ] = b[ t ] + a[ i+1 ]

然后判断n是否在这个数列里面

如果在，那么先手必败。否则不停的减掉数列a中的项构造出n的分解，最后一位就是了。

以上转自神牛：http://hi.baidu.com/lccycc\_acm/item/a6f0dd0ec5c44a39f3eafcd3

做一些解释：首先是a[i]=b[i-1]+1;   b[i-1]是由a[0]……a[i-1]组成的最大的数，那么b[i-1]+1不可能用a[0]……a[i-1]组成。

然后是：if(a[j]\*k<a[i]) b[i]=b[j]+a[i]; else b[i]=a[i];

要求b[j]，表示a[0]……a[i]组成，那么显然是要用到a[i]的，不然不就成了b[i-1]，既然用了a[i]，但是又要使相邻的倍数在K以上。则找到最大的j，使

a[j]\*k<a[i]那么满足条件，便是a[0]……a[j]能组成的最大的数，加上a[i]，那么后者表示当前项不能和之前项组合，那么最大的数就只能是本身

#include<iostream>

#include<cstdio>

#include<cstring>

#define N 2000000

using namespace std;

int a[N],b[N];

int n,k,t,cas=0;

int main(){

scanf("%d",&t);

while(t--){

scanf("%d%d",&n,&k);

int i=0,j=0;

a[0]=b[0]=1;

while(a[i]<n){

i++;

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

while(a[j+1]\*k<a[i])

j++;

if(a[j]\*k<a[i])

b[i]=b[j]+a[i];

else

b[i]=a[i];

}

printf("Case %d: ",++cas);

if(a[i]==n)

puts("lose");

else{

int ans;

while(n){

if(n>=a[i]){

n-=a[i];

ans=a[i];

}

i--;

}

printf("%d\n",ans);

}

}

return 0;

}