

Codeforces Round #654 (Div. 2)

[比赛链接](#)

E2. Asterism (Hard Version)

题意

给定 n 个正整数，代表 n 个敌人，每个敌人有 a_i 颗糖，再给定一个素数 $p(p \leq n)$

定义游戏规则为一开始玩家拥有若干颗糖，每次玩家可以选取一个未挑战且手上糖果数不大于玩家的敌人，打败他获得一颗糖果。

定义 $f(x)$ 为玩家一开始拥有 x 颗糖时可以战胜所有敌人的方案数。

要求输出所有满足 $p \mid f(x)$ 的 x

题解

设 b_i 为糖果数不大于 i 的敌人个数 $C_i(x) = b_{x+i} - i$

有 $f(x) = \prod_{i=0}^{n-1} C_i(x)$

易知 $C_{n-1}(x) \leq 1$ 且 $C_i(x) - C_{i-1}(x) = b_{x+i} - 1 \geq -1$

所以 $p \mid f(x) \iff \forall i (0 \leq i < n \rightarrow C_i \equiv 0 \pmod p)$

由于 $C_i(x) \leq C_{i+1}(x)$ 所以答案为一段区间。

考虑 x 的可能范围，记 $A = \max a_i$ 若 $x \leq A - n$ 则 $f(x) = 0$ 有 $p \mid f(x)$ 若 $x \geq A$ 则 $f(x) = n!$ 有 $p \mid f(x)$

所以将区间左端点初值设为 $A - n + 1$ 右端点初值设为 A 暴力枚举 $i = 0 \sim n - 1$ 的情况，维护一下即可，时间复杂度 $O(n)$

```
#include <cstdio>
#include <iostream>
#include <vector>
#include <algorithm>
#include <cstring>
#include <cctype>
#include <queue>
#include <cmath>
#define _for(i,a,b) for(int i=(a);i<(b);++i)
#define _rep(i,a,b) for(int i=(a);i<=(b);++i)
#define mem(a,b) memset((a),(b),sizeof(a))
using namespace std;
```

```
typedef long long LL;
inline int read_int(){
    int t=0;bool sign=false;char c=getchar();
    while(!isdigit(c)){sign|=c=='-';c=getchar();}
    while(isdigit(c)){t=(t<<1)+(t<<3)+(c&15);c=getchar();}
    return sign?-t:t;
}
inline LL read_LL(){
    LL t=0;bool sign=false;char c=getchar();
    while(!isdigit(c)){sign|=c=='-';c=getchar();}
    while(isdigit(c)){t=(t<<1)+(t<<3)+(c&15);c=getchar();}
    return sign?-t:t;
}
inline void write(LL x){
    register char c[21],len=0;
    if(!x)return putchar('0'),void();
    if(x<0)x=-x,putchar('-');
    while(x)c[++len]=x%10,x/=10;
    while(len)putchar(c[len--]+48);
}
inline void space(LL x){write(x),putchar(' ');}
inline void enter(LL x){write(x),putchar('\n');}
const int MAXN=1e5+5;
int a[MAXN],b[MAXN<<1];
int main()
{
    int n=read_int(),p=read_int(),A=0;
    _for(i,0,n){
        a[i]=read_int();
        A=max(A,a[i]);
    }
    _for(i,0,n)
    b[max(0,a[i]-(A-n))]++;
    _for(i,1,n<<1)
    b[i]+=b[i-1];
    int st=1,en=n;
    _rep(i,1,n){
        while(b[st+(i-1)]<i)
            st++;
    }
    _for(i,0,n){
        while(b[en+i]-i>=p)
            en--;
    }
    st+=(A-n),en+=(A-n);
    if(st<=en){
        enter(en-st+1);
        _rep(i,st,en){
            if(i!=st)
                putchar(' ');
        }
    }
}
```

```

        write(i);
    }
}
else
enter(0);
return 0;
}

```

题解 \$2\$

$$f(x) = \prod_{i=0}^{n-1} C_i(x) = \prod_{i=0}^{n-1} b_{x+i} - i = \prod_{i=x}^{x+n-1} b_{i-x} - i + x$$

所以 $f(x) \equiv 0 \pmod{m}$ 对所有 $x \in [0, n-1]$ 有 x 与 $i - b_{i-x}$ 不同余 $\iff x - (A-n)$ 与 $i - (A-n) - b_{i-x}$ 不同余。

暴力枚举 $x \in [0, A]$ 考虑 $i - b_{i-x}$ 对于 x 与 $x-1$ 的约束只用一项不同，所以可以用滑动窗口维护。

```

#include <cstdio>
#include <iostream>
#include <vector>
#include <algorithm>
#include <cstring>
#include <cctype>
#include <queue>
#include <cmath>
#define _for(i,a,b) for(int i=(a);i<(b);++i)
#define _rep(i,a,b) for(int i=(a);i<=(b);++i)
#define mem(a,b) memset((a),(b),sizeof(a))
using namespace std;
typedef long long LL;
inline int read_int(){
    int t=0;bool sign=false;char c=getchar();
    while(!isdigit(c)){sign|=c=='-';c=getchar();}
    while(isdigit(c)){t=(t<<1)+(t<<3)+(c&15);c=getchar();}
    return sign?-t:t;
}
inline LL read_LL(){
    LL t=0;bool sign=false;char c=getchar();
    while(!isdigit(c)){sign|=c=='-';c=getchar();}
    while(isdigit(c)){t=(t<<1)+(t<<3)+(c&15);c=getchar();}
    return sign?-t:t;
}
inline void write(LL x){
    register char c[21],len=0;
    if(!x)return putchar('0'),void();
    if(x<0)x=-x,putchar('-');
    while(x)c[++len]=x%10,x/=10;
    while(len)putchar(c[len--]+48);
}

```

```
}
inline void space(LL x){write(x),putchar(' ');}
inline void enter(LL x){write(x),putchar('\n');}
const int MAXN=1e5+5;
int a[MAXN],b[MAXN<<1],c[MAXN],ans[MAXN],cnt;
int mod(int a,int p){return (a%p+p)%p;}
int main()
{
    int n=read_int(),p=read_int(),A=0;
    _for(i,0,n){
        a[i]=read_int();
        A=max(A,a[i]);
    }
    _for(i,0,n)
    b[max(0,a[i]-(A-n))]+=;
    _for(i,1,n<<1)
    b[i]+=b[i-1];
    _for(i,0,n)
    c[mod(i-b[i],p)]++;
    _for(i,1,n){
        c[mod(i-1-b[i-1],p)]--;
        c[mod(i+n-1-b[i+n-1],p)]++;
        if(!c[i%p])
            ans[cnt++]=i+A-n;
    }
    enter(cnt);
    _for(i,0,cnt){
        if(i)
            putchar(' ');
        write(ans[i]);
    }
    return 0;
}
```

F. Raging Thunder

题意

题解

From:

<https://wiki.cvbbacm.com/> - **CVBB ACM Team**

Permanent link:

https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:legal_string:jxm2001:contest:cf_654_div_2&rev=1593665405 

Last update: **2020/07/02 12:50**