# CodeChef February Challenge 2021

#### 比赛链接

# **Multiple Games**

## 题意

给定严格递增的正整数序列 \$A\_1,A\_2\cdots A\_n\$□保证 \$A\_i+A\_1\ge A\_{i+1}\$□一开始由我方选定一个 \$G\$□使得 \$0\le G\le M\$□

问必胜场次最多有几场。

### 颞解

首先给出两个博弈游戏等价的定义:对同一个状态(本题为当前石头数),两个博弈游戏要么都是必胜状态要么都是必败状态。

另外假设每次可以拿的石头为 \$[I,r]\$ 个,则必胜状态为 \$G\bmod (I+r)\ge I\$□

#### 接下来给出两个条件:

- 每次可以拿 \$S=\{a\_1,a\_2\cdots a\_k\}\$ 个石头的游戏等价于每次可以拿 \$[\min S,\max S]\$ 个石头的游戏。
- 2. 对任意 \$a\_i,a\_j\in S\$[]若 \$a\_i+\min S\lt a\_j\$[]则存在 \$a\_k\in S\$[]使得 \$a\_i\lt a\_k\lt a\_j\$[]

当条件一成立时,假设存在 \$a\_i+a\_1\lt a\_j\$[]且不存在 \$a\_k\in S\$[]使得 \$a\_i\lt a\_k\lt a\_j\$ 的情况。

于是有 \$j=i+1\$□即 \$a\_i+a\_1\lt a\_{i+1}\$□取 \$G\bmod (a\_1+a\_k)=a\_1+a\_i\$□根据条件一 \$G\$ 是必胜状态。

于是如果选取 \$a 1\sim a i\$□则 \$a 1\le G'\bmod (a 1+a k)\le a I\$□根据条件一 \$G'\$ 是必胜状态。

如果选取 \$a\_{i+1}\sim a\_k\$[]则 \$G'\bmod (a\_1+a\_k)\gt 2a\_1+a\_i\$[]根据条件一 \$G'\$ 是必胜状态。

于是 \$G\$ 是必败状态,矛盾。于是充分性证毕。

当条件二成立时,首先考虑 \$0\le G\lt a\_1+a\_k\$□易知 \$0\lt G\lt a\_1\$ 是必败状态。

当 \$a\_i\le G\lt a\_{i+1}\$ 时,取 \$a\_i\$ 个石头,根据条件二,有 \$a\_i+a\_1\ge a\_{i+1}\$□于是 \$G'=G-a\_i\lt a\_1\$ 是必败状态。

于是 \$a\_1\le G\lt a\_k\$ 是必胜状态。当 \$a\_k\le G\lt a\_1+a\_k\$ 时取 \$a\_k\$ 个石头有 \$G'\lt a\_i\$□于是 \$G\$ 也是必胜状态。

于是 \$0\le G\lt a 1+a k\$ 时必胜状态为 \$a 1\le G\lt a 1+a k\$[]

数学归纳法设 \$k(a 1+a k)\le G\lt (k+1)(a 1+a k)\$ 满足条件一。

当 \$(k+1)(a\_1+a\_k)\le G\lt (k+1)(a\_1+a\_k)+a\_1\$ 时,任意取石头 \$a\_1\sim a\_k\$[]

发现总有 \$k(a\_1+a\_k)+a\_1\le G'\lt (k+1)(a\_1+a\_k)\$ 全是必胜状态,于是 \$G\$ 是必败状态。

当 \$(k+1)(a\_1+a\_k)+a\_1\le G\lt (k+2)(a\_1+a\_k)\$ 类比 \$a\_1\le G\lt a\_1+a\_k\$ 的取法即可到达必败状态,于是 \$G\$ 是必胜状态。必要性证毕。

回到原题,现在只需要考虑选取 \$G\$ 使得其满足尽可能多的 \$G\bmod (a\_{l\_i}+a\_{r\_i})\ge a\_{l\_i}\$ 即可。

考虑维护  $$0\le G\le m$$  的答案数组。对  $a_{l_i}+a_{r_i}\le sqrt m$$  的询问,可以转化为不超过 sO(sqrt m) 次区间加操作。

利用差分和前缀和可以 \$O(\sqrt m)\$ 处理每个上面询问。

对  $a_{l_i}+a_{r_i}\leq m$  的询问,考虑用  $o(\sqrt m)$  个长度不超过  $o(\sqrt m)$  的数组 c 维护贡献。

对每个上面询问使得 \$c(l\_i+r\_i)(l\_i\sim r\_i)\$ 加一。

最后从 \$0\sim m\$ 扫描一遍答案数组,同时加上这 \$O(\sqrt m)\$ 的数组的当前位置贡献,然后每个数组指针移动一位。

总时间复杂度 \$O((m+q)\sqrt m)\$□

```
const int MAXN=2e5+5,MAXM=500;
int a[MAXN],s[MAXN],c[MAXM][MAXM],p[MAXM];
int main()
    int T=read int();
    while(T--){
        int n=read int(),q=read int(),m=read int(),blk=sqrt(m)+1;
        rep(i,0,m)s[i]=0;
        for(i,1,blk){
            _for(j,0,i)c[i][j]=0;
            p[i]=0;
         rep(i,1,n)a[i]=read int();
        while(q--){
            int l=read int(),r=read int();
            if(a[l]+a[r]>=blk){
                 int pos=a[l];
                 while(pos<=m) {</pre>
                     s[pos]++;
                     if(pos+a[r]<=m)s[pos+a[r]]--;</pre>
                     pos+=a[l]+a[r];
                 }
```

https://wiki.cvbbacm.com/ Printed on 2025/11/29 16:08

From:

https://wiki.cvbbacm.com/ - CVBB ACM Team

Permanent link:

 $https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:legal\_string:jxm2001:contest:cf\_feb21\&rev=1613392692$ 

Last update: 2021/02/15 20:38