Comet oj Contest #3题解

A.比赛

出题人**NiroBC** 暴力列出所有可能的比赛，排序后取前K大。

热心网友**晚来风雪水长东**

题目上说总共有n只小猫，每只小猫的实力值是a_i，每两只小猫之间会进行一场精彩度为a_i+a_j的比赛，所以我们可以用两层for循环遍历每两只小猫来将每两只小猫比赛的精彩度全部存入num数组，再对num数组进行从小到大$sort$排序，因为num数组最后的那项是最大的，因此从最后那项开始往前加，加够k个就是前k大之和了，即为答案。

参考代码

B.棋盘

出题人**NiroBC**

状压dp。

热心网友**晚来风雪水长东**

题目意思就是说通过把0变成1来将所有1变成一个联通块，要求最少的变换个数。我是用的暴力模拟，从第一列开始遍历每一列，因为只有两行，含1的每一列无非就是：上1下1，上1下0，上0下1；只有这三种情况是需要变化前面列的0来使1联通的，观察这三种情况，上1下1时，变换个数就是上一次有1的列号与当前列号差值的绝对值-1(直接由上一个有1的位置沿着行直线变化到当前列)；上1下0时，就需要看上一次有1的那一列的1的情况了，如果上一次有1处和当前列有1处是同一行，说明可以直线变换过来，变换个数就是上一次有1的列号与当前列号差值的绝对值-1，如果不是同一行，则需要多加1（需要多变换一个棋子）；上0下1的情况和上一种情况类似解法。因此关键就是保存上一次有1处的那列情况，在遇到某列有1时考虑上面的情况然后累加就行了。

参考代码

C.子序列子序列子序列...

出题人**ccz181078**

设$m=m_0 2^c$，其中m_0是奇数。 若一个序列$a_1,a_2,\ldots,a_k$包含k个元素，则它的每个元素出现在2^{k-1}个子序列中，因此它是完美的当且仅当$2^{k-1}(a_1+a_2+\ldots+a_k)$是$m$的倍数，即$(a_1+a_2+\ldots+a_k)$是$m/2^{min(k-1,c)}$的倍数。 记$f(i,j,k)$表示$a_1,a_2,\ldots,a_i$中，长度$j$ $(1\le j\le c)$，求和后对$m/2^{j-1}$取模得到k的子序列的个数；$g(i,k)$表示$a_1,a_2,\ldots,a_i$中，长度至少为$c+1$，求和后对$m/2^c$取模得到k的子序列的个数。考虑一个子序列是否包含a_i，可以递推求出$i=n$时的结果，总的答案就是$g(n,0)+\sum_{j=1}^{c} f(n,j,0)$。 计算$f(i,j,k)$是$O(1)$的，而对每个$j$，$f(i,j,k)$共有$nm/2^{j-1}$项，因此总的时间复杂度是$O(nm(1+1/2+1/4+\ldots))=O(nm)$。

D.可爱的菜菜子

出题人**nzhtl1477**

如果是单点修改，这题就是一个线段树维护区间线性基的题，但这里是区间xor

我们考虑维护原序列的xor差分，这样每次修改的时候只需要修改两个位置的值就可以了，不用修改一个区间。查询的时候，只需要特判边界，因为简单的线性代数知识告诉我们维护xor差分的线性基和原序列的线性基是等价的。

总复杂度$O(nlognlog^2w)$，似乎存在$O(nlognlogw)$的四毛子做法。

E.说好的交换律呢?

出题人**dreamoon**

最暴力的有了这个式子，我们已经能做到 $O(x)$ 了！

但若只是这样，这题就不会是$problem E$ 的难度。

$x \ / \ i * j = x * j \ / \ i$ 这个条件很难去解析，所以先试着转换这个条件：

(虽然叙述里的 $/$ 号是整数除法，但从这里开始，使用的除号都是精确运算。) (在这里 x % i 的运算符和 C 语言意思一样，都代表着整数 x 除以整数 i 的余数，值为一个介于 $0 \sim i-1$ 之间的数。)

叙述里的条件就是:

$(x - x \ \% \ i) \ / \ i * j = (x * j - (x * j) \ \% \ i) \ / \ i$

首先，左右同乘 i 后得到：

$(x - x \ \% \ i) * j = x * j - (x * j) \ \% \ i$

左右两式减去 $x * j$ 并取负后得到：

$x \ \% \ i * j = (x * j) \ \% \ i$

其中 $(x * j) \ \% \ i$ 就是 $((x \ \% \ i) * j) \ \% \ i$

令 $v = (x \ \% \ i) * j$

至此我们推得 $v = v \ \% \ i$。

$v \ \% \ i$ 的充要条件就是 $v < i$

所以最终我们得到 $(x\%i) * j < i \rightarrow j < \frac{i}{x \ \% \ i}$。

若固定 i 的值，我们能推论出 $1 \leq j \leq \lceil \frac{i}{x \ \% \ i} \rceil - 1$。

于是本题答案为：$\sum_{i=1}^{a} \min(\lceil \frac{i}{x \ \% \ i} \rceil - 1, b)$

于是我们已经能做到 $O(x)$ 了！

但若只是这样，这题就不会是 $problem E$ 的难度。

接着，我们分成 $i > x$ 和 $i \leq x$ 的情况。

$i > x$ 时，$x \ \% \ i$ 的值就是 x，

可能组数为 $\sum_{i=x+1}^{a} (\lceil \frac{i}{x} \rceil - 1) = x \cdot \sum_{i=1}^{\lfloor a/x \rfloor - 1} i + (a \ \% \ x) \ \cdot \lfloor \frac{a}{x} \rfloor$。

可以 $O(1)$ 算出。

$i \leq x$ 时，其作法和以下经典问题类似：

> 给定 n，求 $\sum_{i=1}^{n} \lfloor \frac{n}{i} \rfloor$。

这个经典问题的解法大概就是：$\lfloor \frac{n}{i} \rfloor$ 的可能值只有 $O(\sqrt{n})$ 个，且对于每个可能的 $\lfloor \frac{n}{i} \rfloor$ 值，能得到该值的 i 都是连续的整数。

拿 $n=100$ 时做举例 对于 $i=1\sim 100$，$\lfloor \frac{n}{i} \rfloor$ 的值分别是：(连续一样的值就用中括号括起来)

$[100],[50],[33],[25],[20],[16],[14],[12],[11],[10],[9],[8],[7,7],[6,6],[5,5,5,5],[4,4,4,4,4],[3,3,3,3,3,3,3,3]$,
[2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2],
$[1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1]$

并且能够计算出，当 $\lfloor \frac{n}{i} \rfloor = c$ 时，i 的可能值的上界为：$\lfloor \frac{n}{c} \rfloor$，于是我们就能用下列代码 $O(\sqrt{n})$ 算出所有数的答案。

```
#include<bits/stdc++.h>
long long f(long long n){
    long long i=1,ret=0;
    while(i<=n){
        long long v=n/i;
        long long nxt_i=x/v; //nxt_i = 最大的 x 满足 n/x 和 n/i 一样
        ret+=(nxt_i+1-i)*v;
        i=nxt_i+1;
    }
    return ret;
}
int main(){
    long long n;
    scanf("%lld",&n);
    solve(n);
    return 0;
}
```

现在我们要用相同概念计算本问题的答案。

这里不妨假设 $a=b=x$ (a,b 不等于 x 的情况请自己微调算式吧)。

本题答案在这种情况下是：$\sum_{i=1}^{x}(\lceil \frac{i}{x\ \%\ i} \rceil - 1)$

我们这次同样对于不同的 $\lfloor \frac{x}{i} \rfloor = c$ 的值来分开枚举，当 $\lfloor \frac{x}{i} \rfloor = c$ 时，我们要再把 i 根据不同的 $\lceil \frac{i}{x\ \%\ i} \rceil$ 值做枚举，当 $\lfloor \frac{x}{i} \rfloor$ 和 $\lceil \frac{i}{x\ \%\ i} \rceil$ 的值固定时，可能的 i 值也是连续的！

例如 $x=100$ 时，把 $i=1\sim 100$ 的数对 $(\lfloor \frac{x}{i} \rfloor, \lceil \frac{i}{x\ \%\ i} \rceil)$ 写下来就是：(当 i 整除 n 时，下表记 $\lceil \frac{i}{x\ \%\ i} \rceil = 100$ 。)

$[(100,100)],[(50,100)],[(33,3)],[(25,100)],[(20,100)],[(16,1)],[(14,3)],[(12,2)],[(11,9)],[(10,100)],[(9,11)]$,
$[(8,3)],[(7,1)],[(7,7)]$,
$[(6,1)],[(6,4)],[(5,1),(5,1)],[(5,3)],[(5,100)],[(4,1),(4,1)],[(4,2)],[(4,6)],[(4,100)]$,
$[(3,1),(3,1),(3,1)],[(3,2)],[(3,3)],[(3,4)],[(3,8)],[(3,33)],[(2,1),(2,1),(2,1),(2,1),(2,1),(2,1)]$,
$[(2,2),(2,2),(2,2)],[(2,3),(2,3)],[(2,4)],[(2,5)],[(2,7)],[(2,12)],[(2,24)],[(2,100)]$,
$[(1,1),(1,1),(1,1),(1,1),(1,1),(1,1),(1,1),(1,1),(1,1),(1,1),(1,1),(1,1),(1,1),(1,1),(1,1),(1,1)]$,
$[(1,2),(1,2),(1,2),(1,2),(1,2),(1,2),(1,2),(1,2)],[(1,3),(1,3),(1,3),(1,3),(1,3)]$,
$[(1,4),(1,4),(1,4),(1,4)],[(1,5),(1,5)],[(1,6),(1,6)],[(1,7)],[(1,8)],[(1,9)],[(1,10)],[(1,11)]$,
$[(1,13)],[(1,15)],[(1,19)],[(1,24)],[(1,32)],[(1,49)],[(1,99)],[(1,100)]$

如上，$n=100$ 时，总共有 57 个区间。

虽然这个时候 n 的值和区间数比例看起来差很小，但实际上，当 $n=10^9$ 时，区间数为 693197，数量很少的，有数学办法能证明，区间数量为 $O(\sqrt{n}\cdot\log(n))$ (最后面会补充)。

对于一组 $(\lfloor \frac{x}{i} \rfloor, \lceil \frac{i}{x\ \%\ i} \rceil)$，上界可以如下方式计算：

令 $(\lfloor \frac{x}{i} \rfloor, \lceil \frac{i}{x \ \% \ i} \rceil)$ 的值等于 (c, q)，同时 q 也等于 $\lceil \frac{i}{x-c \cdot i} \rceil$，于是可发现，当 c 为定值时，$\frac{x}{x \ \% \ i}$ 的值随 i 值递增而递增。于是上界就是满足 $\lceil \frac{i}{x-c \cdot i} \rceil \leq q$ 的最大的 i。

$\lceil \frac{i}{x-c \cdot i} \rceil \leq q \rightarrow \frac{i}{x-c \cdot i} \leq q+1 \rightarrow i \leq (q+1) \cdot (x-c \cdot i) \rightarrow i \cdot (c+c \cdot q+1) \leq (q+1) \cdot x$

$\rightarrow i \leq \frac{(q+1) \cdot x}{c+c \cdot q+1}$

所以上界会是 $\lfloor \frac{(q+1) \cdot x}{c+c \cdot q+1} \rfloor$。

最后是要描述下区间数量的复杂度证明，这是本题最难的部分了，相信很多人都和出题者本人一样，代码先写再说，复杂度的证明就先搁着(实际上此证明试验题者给出的...，比出题者出题时约略证出的复杂度还优秀，orz ccz181078)。

证明如下：

再提示大家一些变数的定义一次：假定 $\lfloor \frac{x}{i} \rfloor, \lceil \frac{i}{x \ \% \ i} \rceil$ 的值分别是 c, q。

先假定有一个常数 M。

对于相同一个 c 而言，

q 值 $\leq M$ 的组数最多有 q 组。

q 值 $> M$ 时，

1. 由于 $q = \lceil \frac{i}{x-c \cdot i} \rceil > M \rightarrow \lfloor \frac{i}{x-c \cdot i} \rfloor \geq M \rightarrow \frac{i}{x-c \cdot i} \geq M \rightarrow i > M \cdot (x-c \cdot i)$
   $\rightarrow (1+M \cdot c) \cdot i \geq M \cdot x \rightarrow i \geq \frac{M \cdot x}{1+M \cdot c} \rightarrow i \geq \frac{x}{\frac{1}{M}+c}$
2. 由于 $\lfloor \frac{x}{i} \rfloor = c$，$\rightarrow \frac{x}{i} \geq c \rightarrow i \leq \frac{x}{c}$

综合和 1.2.，我们有 $\frac{x}{\frac{1}{M}+c} \leq i \leq \frac{x}{c}$，所以 i 的可能数量至多约有 $\frac{x}{c} - \frac{x}{\frac{1}{M}+c} = \frac{\frac{1}{M} \cdot x}{c(\frac{1}{M}+c)} = \frac{x}{1+c^2 \cdot M} \leq \frac{x}{c^2 \cdot M}$。

最后我们取 $M = \frac{\sqrt{x}}{c}$，那么可证明区间数量不超过 $M + \frac{x}{c^2 \cdot M} = \frac{\sqrt{x}}{c} + \frac{\sqrt{x}}{c} = O(\frac{\sqrt{x}}{c})$

最后按不同的 c 值做加总能得到区间数的复杂度为：

$O(\sum_{c=1}^{x} \frac{\sqrt{x}}{c}) = O(\sqrt{x} \cdot \sum_{k=1}^{x} \frac{1}{k}) = O(\sqrt{x} \cdot \ln x)$

F.毒瘤的菜菜子

出题人**nzhtl1477**

第十四分块

我们考虑莫队维护

每次把区间$[l,r]$拓展到$[l,r+1]$的时候需要维护$a[r+1]$和$[l,r]$这段区间的贡献

这里利用莫队二次离线的$trick$：

莫队就可以抽象为$O(nsqrtm)$次转移，每次查询一个序列中的位置对一个区间$[l,r]$的贡献

这个贡献是可以差分的， 也就是说把每次转移的$a[r+1]$对区间$[l,r]$的贡献差分为$a[r+1]$对前缀$[1,r]$的贡献减去对前缀$[1,l-1]$的贡献

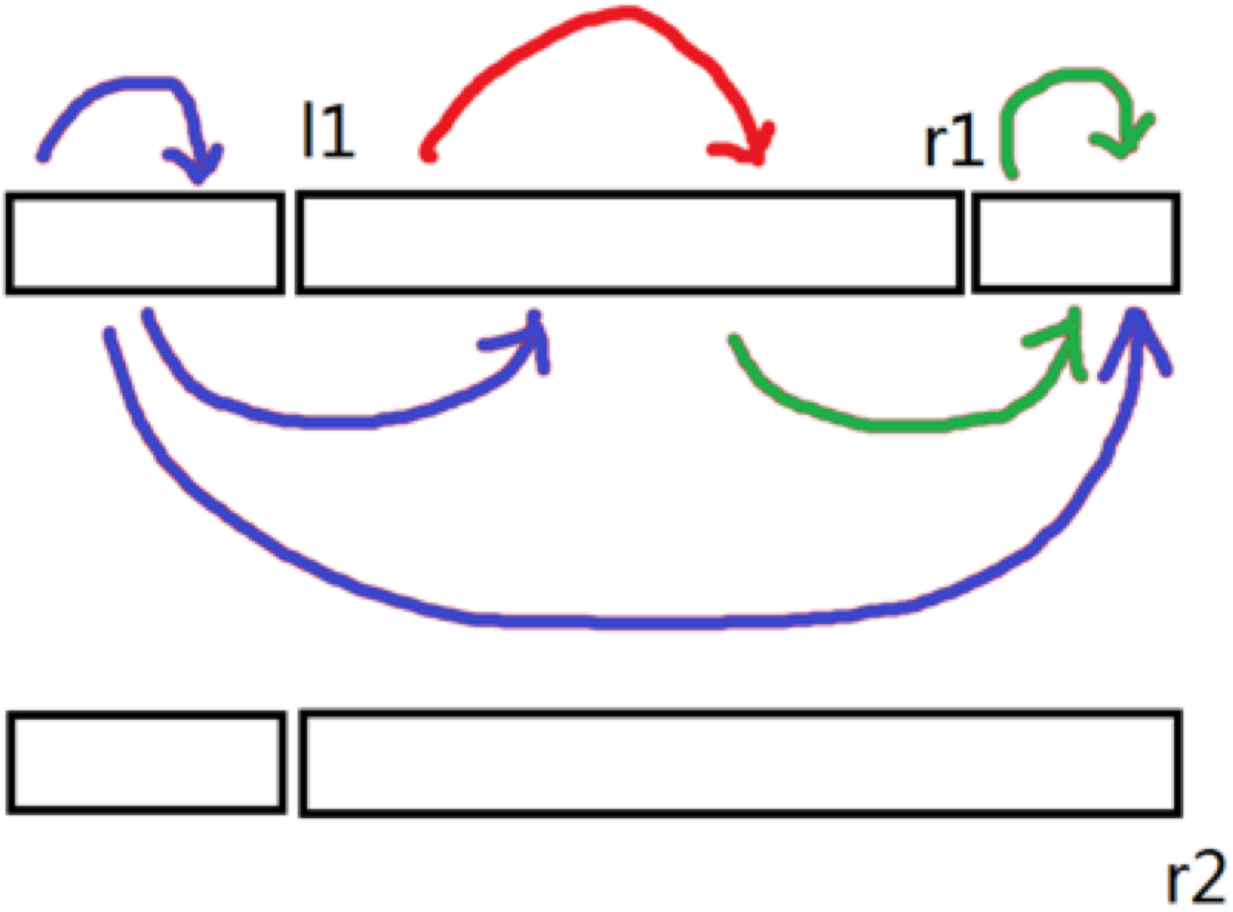
然后通过这个差分我们可以发现，我们把$O(nsqrtm)$次的修改降低到了$O(n)$次修改,因为前缀只会拓展$O(n)$次

于是我们每次可以较高复杂度拓展前缀和，因为插入次数变成了$O(n)$次

然后这里我们离线莫队的转移的时候可以做到线性空间，因为每次莫队是从一个区间转移到另一个区间：

我们记下pre[i]为i的前缀区间的答案，$suf[i]$为i的后缀区间的答案

考虑把区间$[l1,r1]$拓展到区间$[l1,r2]$:



$[l1,r1]$区间的答案为红色部分的贡献，$[l1,r2]$区间的答案为红色部分加上绿色部分的贡献，$r2$的前缀贡献为所有的贡献，$r1$的前缀贡献为红色的贡献与前两个蓝色的贡献

于是我们可以把绿色的贡献，也就是两次询问的差量差分为$pre[r2]-pre[r1]-[1,l1-1]$对$[r1+1,r2]$的贡献

然后我们再差分一下$[1,l1-1]$对$[r1+1,r2]$的贡献为$[1,l1-1]$对$[1,r2]$的贡献减去$[1,l1-1]$对$[1,r1]$的贡献，这里每个位置开个$vector$4存一下就可以了

于是我们达成了$O(n+m)$的空间复杂读将莫队给二次离线

现在考虑怎么算这个贡献，我们要高效计算区间$[l,r]$对x的贡献时

先把贡献拆开变成：$[l,r]$中x的倍数个数，$[l,r]$中x的约数个数

$[l,r]$中x的倍数个数：每次前缀加入一个数$a[i]$的时候，我们把$a[i]$的约数位置所对应的数组$pre1$的位置都$++$然后查询的时候直接查$pre1[x]$即可得到x倍数个数

$[l,r]$中x的约数个数：这里考虑对x的约数进行一次根号分治

对于所有x大于$sqrtn$的约数，我们拓展前缀的时候维护，即每次把$a[i]$的倍数位置所对应的数组$pre2$的位置都$++$然后查询的时候直接查$pre2[x]$即可得到$x_{大于sqrt}$的约数个数

对于所有x小于$sqrtn$的约数，我们考虑每次直接枚举区间$[l,r]$中$1,2,\ldots,sqrtn$的出现次数

这里可以用$fractionalcascading$的$trick$做到$O(sqrtn+logn)$单次，线性空间

于是我们得到了一个总时间复杂度$O(nsqrtm)$，空间$O(n+m)$的做法