离线算法杂题选讲

杨明天

浙江省镇海中学

2018 年 12 月 28 日

mex

题目大意

给定一个长度为n的数列A。m次询问，每次询问$\operatorname{mex} A_{l\sim r}$。
$n, m \leq 2 \times 10^5; 0 \leq A_i \leq 10^9$

题目来源

BZOJ 3585

mex

Solution

mex
Solution

显然$\geq n$的数是没用的。

mex

Solution

显然$\geq n$的数是没用的。
对于$< n$的数，维护每个数出现的次数$cnt[x]$。

mex

Solution

显然$\geq n$的数是没用的。
对于$< n$的数，维护每个数出现的次数$cnt[x]$。
考虑询问区间的移动对答案mex的影响。

mex
Solution

显然$\geq n$的数是没用的。
对于$< n$的数，维护每个数出现的次数$cnt[x]$。
考虑询问区间的移动对答案mex的影响。
加入数x后，若$cnt[\mathrm{mex}] \neq 0$，则暴力计算新的mex。

mex

Solution

显然$\geq n$的数是没用的。
对于$< n$的数，维护每个数出现的次数$cnt[x]$。
考虑询问区间的移动对答案mex的影响。
加入数x后，若$cnt[\text{mex}] \neq 0$，则暴力计算新的mex。
删除数x后，若$cnt[x] = 0$，则对当前mex取min。

mex

Solution

显然$\geq n$的数是没用的。
对于$< n$的数，维护每个数出现的次数$cnt[x]$。
考虑询问区间的移动对答案mex的影响。
加入数x后，若$cnt[\mathrm{mex}] \neq 0$，则暴力计算新的mex。
删除数x后，若$cnt[x] = 0$，则对当前mex取$\min$。
时间复杂度$\mathcal{O}((n+m)\sqrt{n})$。

Coprimes

题目大意

一个长度为n的数列A，q次询问，每次询问$A_{[l,r]}$中有多少长度为k的子序列，满足序列内所有元素的$\gcd = 1$。
$n, q \leq 5 \times 10^4; 1 \leq A_i \leq 10^5$

题目来源

2015 ACM-ICPC Asia-Amritapuri Site Onsite Round

Coprimes

Solution

对于单个询问，答案为$\sum_{x=1}^{10^5}\mu(x)\binom{k}{cnt[x]}$。其中$cnt[x]$表示$A_{[l,r]}$中$x$倍数的个数。

当x有平方因子时，$\mu(x)=0$，对答案没有贡献。若我们用d表示10^5以内不同的质数个数，则最多有2^d个x需要考虑。

若直接套用莫队算法，时间复杂度$\mathcal{O}(2^d q\sqrt{n})$，还是不能通过本题。

考虑将答案写成如下形式：

$$\sum\binom{y}{k}cntp[y]-\sum\binom{y}{k}cntn[y]$$

其中，$cntp[y]$表示满足$cnt[x]=y$且$\mu(x)=1$的x的个数；$cntn[y]$表示满足$cnt[x]=y$且$\mu(x)=-1$的x的个数。

Coprimes

Solution

注意到满足$cnt[x] \geq \sqrt{n}$最多有$\sqrt{n}$个，因此，当$cnt[x] \geq \sqrt{n}$时，我们可以单独存储，剩下的用$cntp[]$和$cntn[]$来表示。

此时，我们就把贡献分为了$y = cnt[x] < \sqrt{n}$和$cnt[x] \geq \sqrt{n}$两部分。

最后答案可以表示为：

$$\sum \binom{y}{k}(cntp[y] - cntn[y]) + \sum \mu(x)\binom{k}{cnt[x]}$$

时间复杂度$\mathcal{O}(q\sqrt{n})$。

Mokia

题目大意

一个$n \times n$的矩阵，初始时每个格子里的数全为0。m次操作，操作包含以下两种：

- 将某个格子加上一个数；
- 询问某个子矩阵的值。

$n \leq 5 \times 10^5; m \leq 2 \times 10^5$

题目来源

Balkan OI 2007

Mokia

Solution

Mokia

Solution

将所有操作离线，每个询问拆分成4个，修改和询问放在一起分治，树状数组处理，每层分治处理左区间中修改对右区间询问的贡献即可。

Mokia

Solution

将所有操作离线，每个询问拆分成4个，修改和询问放在一起分治，树状数组处理，每层分治处理左区间中修改对右区间询问的贡献即可。

时间复杂度$\mathcal{O}(m\log^2 m)$。

适者

题目大意

有n个敌人，每个敌人有一个攻击力v_i和一个防御力d_i。你的攻击力是a。战斗看做回合制，每回合进程如下：

你选择某个敌人进行攻击，令其防御力减少a，若防御力< 0则该敌人被击败。

所有存活的敌人每人对你造成v_i点损失。

战斗开始前你有机会直接去掉对方的两个敌人。

你拥有无限的血量，请最小化你的损失。

$n \leq 3 \times 10^5$

题目来源

BZOJ 4700

适者

Solution

适者

Solution

首先不考虑去掉两个敌人的情况。

适者

Solution

首先不考虑去掉两个敌人的情况。

显然我们可以按一定顺序依次击败各个敌人，而不必一个人打一半就去打其他人。

适者

Solution

首先不考虑去掉两个敌人的情况。

显然我们可以按一定顺序依次击败各个敌人，而不必一个人打一半就去打其他人。

用t_i表示击败敌人i所花的时间，则$t_i = \lceil \frac{d_i}{a} \rceil$。

适者

Solution

首先不考虑去掉两个敌人的情况。

显然我们可以按一定顺序依次击败各个敌人，而不必一个人打一半就去打其他人。

用t_i表示击败敌人i所花的时间，则$t_i = \lceil \frac{d_i}{a} \rceil$。

考虑攻击的顺序。i在j前面，当且仅当$t_i \times v_j < t_j \times v_i$。

适者

Solution

首先不考虑去掉两个敌人的情况。

显然我们可以按一定顺序依次击败各个敌人，而不必一个人打一半就去打其他人。

用t_i表示击败敌人i所花的时间，则$t_i = \lceil \frac{d_i}{a} \rceil$。

考虑攻击的顺序。i在j前面，当且仅当$t_i \times v_j < t_j \times v_i$。

此时$ans = \sum_{i=1}^{n} (\sum_{j=1}^{i} t_j - 1) \times v_i$。

适者

Solution

适者

Solution

现在考虑事先去掉两个敌人的情况，即如何选择去掉的敌人使损失最小。

适者

Solution

现在考虑事先去掉两个敌人的情况，即如何选择去掉的敌人使损失最小。

用pre表示t_i的前缀和，用suf表示v_i的后缀和。

适者

Solution

现在考虑事先去掉两个敌人的情况，即如何选择去掉的敌人使损失最小。

用pre表示t_i的前缀和，用suf表示v_i的后缀和。

考虑只去掉一个敌人的情况，去掉i这个敌人可以使答案减少$c_i = suf[i] \times t_i + (pre[i-1]-1) \times v_i$。

适者

Solution

现在考虑事先去掉两个敌人的情况，即如何选择去掉的敌人使损失最小。

用pre表示t_i的前缀和，用suf表示v_i的后缀和。

考虑只去掉一个敌人的情况，去掉i这个敌人可以使答案减少$c_i = suf[i] \times t_i + (pre[i-1]-1) \times v_i$。

而去掉两个人就是要找到一对i,j，使得$c_i + c_j - v_i \times t_j$最大。

适者

Solution

现在考虑事先去掉两个敌人的情况，即如何选择去掉的敌人使损失最小。

用pre表示t_i的前缀和，用suf表示v_i的后缀和。

考虑只去掉一个敌人的情况，去掉i这个敌人可以使答案减少$c_i = suf[i] \times t_i + (pre[i-1]-1) \times v_i$。

而去掉两个人就是要找到一对i, j，使得$c_i + c_j - v_i \times t_j$最大。

考虑固定一个i，则j的贡献就是一个关于v_i的一次函数，用李超树或CHT维护凸壳或使用CDQ分治。

适者

Solution

现在考虑事先去掉两个敌人的情况，即如何选择去掉的敌人使损失最小。

用pre表示t_i的前缀和，用suf表示v_i的后缀和。

考虑只去掉一个敌人的情况，去掉i这个敌人可以使答案减少$c_i = suf[i] \times t_i + (pre[i-1] - 1) \times v_i$。

而去掉两个人就是要找到一对i, j，使得$c_i + c_j - v_i \times t_j$最大。

考虑固定一个i，则j的贡献就是一个关于v_i的一次函数，用李超树或CHT维护凸壳或使用CDQ分治。

时间复杂度$\mathcal{O}(n \log n)$。

王老先生

题目大意

有m块田地和n个人，田地编号$1 \sim m$，每块田地都有一个主人。q次操作，第i次操作会让每个拥有编号$[l_i, r_i]$中田地的人赚c_i元。每个人都有一个赚钱的目标值v_i，求每个人是在第几次操作之后达到他的目标的。

$n, m \leq 10^5; c_i, v_i \leq 10^9$

题目来源

TOI 2015 二模 B

王老先生

Solution

王老先生

Solution

对于同一个人管理的不同田地以及一次操作区间$[l,r]$，不妨将贡献全归给落在$[l,r]$中最左边的那个田地。

王老先生

Solution

对于同一个人管理的不同田地以及一次操作区间$[l, r]$，不妨将贡献全归给落在$[l, r]$中最左边的那个田地。

对于某个田i，假设下一个和他同个主人的田在j，那么会使i有贡献的区间$[l, r]$要满足$l \le i$以及$r < j$。

王老先生

Solution

对于同一个人管理的不同田地以及一次操作区间$[l, r]$，不妨将贡献全归给落在$[l, r]$中最左边的那个田地。

对于某个田i，假设下一个和他同个主人的田在j，那么会使i有贡献的区间$[l, r]$要满足$l \leq i$以及$r < j$。

每次二分时相当于在解决一个二维查询修改问题，故将其中一维排序以达成降维。

王老先生

Solution

对于同一个人管理的不同田地以及一次操作区间$[l, r]$，不妨将贡献全归给落在$[l, r]$中最左边的那个田地。

对于某个田i，假设下一个和他同个主人的田在j，那么会使i有贡献的区间$[l, r]$要满足$l \leq i$以及$r < j$。

每次二分时相当于在解决一个二维查询修改问题，故将其中一维排序以达成降维。

对操作区间以及田地贡献区间的右界分别排序即可使用树状数组处理上述二维问题。

王老先生

Solution

对于同一个人管理的不同田地以及一次操作区间$[l, r]$，不妨将贡献全归给落在$[l, r]$中最左边的那个田地。

对于某个田i，假设下一个和他同个主人的田在j，那么会使i有贡献的区间$[l, r]$要满足$l \le i$以及$r < j$。

每次二分时相当于在解决一个二维查询修改问题，故将其中一维排序以达成降维。

对操作区间以及田地贡献区间的右界分别排序即可使用树状数组处理上述二维问题。

因为每次二分是离线的，所以数据结构内的东西不能传给下次二分。不过可将目标扣掉现阶段赚的钱当作新的目标，传给下次二分。

王老先生

Solution

对于同一个人管理的不同田地以及一次操作区间$[l, r]$，不妨将贡献全归给落在$[l, r]$中最左边的那个田地。

对于某个田i，假设下一个和他同个主人的田在j，那么会使i有贡献的区间$[l, r]$要满足$l \le i$以及$r < j$。

每次二分时相当于在解决一个二维查询修改问题，故将其中一维排序以达成降维。

对操作区间以及田地贡献区间的右界分别排序即可使用树状数组处理上述二维问题。

因为每次二分是离线的，所以数据结构内的东西不能传给下次二分。不过可将目标扣掉现阶段赚的钱当作新的目标，传给下次二分。

时间复杂度$\mathcal{O}(n + q \log n \log q)$。

Thanks

谢谢大家