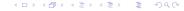
良心温暖信心赛题解

老K

2018年11月2日



前言



■ CF1043F





- CF1043F
- Tag: 数论, 状压 DP







■ CF1043F

■ Tag: 数论, 状压 DP

■ 定位: D2T2



前言

CF1043F

■ Tag: 数论, 状压 DP

■ 定位: D2T2

真实模拟联赛数据强度。

算法 1

cout (a[1]==1?1:-1) (endl;



- cout (a[1]==1?1:-1) (endl;
- 期望得分 1。



算法 2

■ 爆搜每种情况判断。



Conscience

O
OOOO

- 爆搜每种情况判断。
- 期望得分 20。

题解

■ 设 dp; 为最小的 gcd 为 i 的子集大小。



- 设 *dp_i* 为最小的 gcd 为 *i* 的子集大小。
- 直接 $O(n^2)$ 转移 (枚举 j, k > i 并且 gcd(j, k) = i)。

- 设 dp; 为最小的 gcd 为 i 的子集大小。
- 直接 $O(n^2)$ 转移 (枚举 j, k > i 并且 gcd(j, k) = i)。
- 时间复杂度 $O(n^3)$ 。

- 设 *dp_i* 为最小的 gcd 为 *i* 的子集大小。
- 直接 $O(n^2)$ 转移 (枚举 j, k > i 并且 gcd(j, k) = i)。
- 时间复杂度 O(n³)。
- 期望得分 15(结合算法 2 期望得分 35)

■ 事实上枚举的 j, k 如果不满足 i lj, k 显然不可能,那么 j, k 只需要枚举 j 的倍数。

- 事实上枚举的 j, k 如果不满足 i|j, k 显然不可能,那么 j, k 只需要枚举 i 的倍数。
- 时间复杂度 O(n²)?

- 事实上枚举的 j, k 如果不满足 i|j, k 显然不可能,那么 i, k 只需要枚举 i 的倍数。
- 时间复杂度 O(n²)?
- 期望得分 30(结合算法 2 期望得分 50)

算法 5

■ 首先考虑把 ai 表示为一个二进制数: 第 i 位表示是否 是第i个因子的倍数。那么满足条件的方案一定满足 and 和为 0

- 首先考虑把 a; 表示为一个二进制数: 第 i 位表示是否 是第i个因子的倍数。那么满足条件的方案一定满足 and 和为 0
- 由于 $a_i \le 300000$, 所以 a_i 的不同质因子个数 < 7。



- 首先考虑把 a; 表示为一个二进制数: 第 i 位表示是否 是第i个因子的倍数。那么满足条件的方案一定满足 and 和为 0
- 由于 $a_i < 300000$,所以 a_i 的不同质因子个数 < 7。
- 假设已经固定选择 a_i, 那么就只需要考虑这 7 个 1 要 变成 0



- 首先考虑把 a; 表示为一个二进制数: 第 i 位表示是否 是第i个因子的倍数。那么满足条件的方案一定满足 and 和为 0
- 由于 $a_i < 300000$,所以 a_i 的不同质因子个数 < 7。
- 假设已经固定选择 a_i, 那么就只需要考虑这 7 个 1 要 变成 0
- 通过容斥计算出这几位的每种情况是否存在,然后 dp 即可。



- 首先考虑把 a; 表示为一个二进制数: 第 i 位表示是否 是第i个因子的倍数。那么满足条件的方案一定满足 and 和为 0
- 由于 $a_i < 300000$,所以 a_i 的不同质因子个数 < 7。
- 假设已经固定选择 a_i, 那么就只需要考虑这 7 个 1 要 变成 0
- 通过容斥计算出这几位的每种情况是否存在,然后 dp 即可。
- 技巧比较多?







csa sortall.





- csa sortall.
- 定位: D2T3

- csa sortall.
- 定位: D2T3
- tag: 树套树。

- csa sortall.
- 定位: D2T3
- tag: 树套树。
- 简化题意:

前言

- csa sortall.
- 定位: D2T3
- tag: 树套树。
- 简化题意:
- 对于一个区间,快乐度定义为区间升序排序离散化后, 每个数乘上序号的和。

前言

- csa sortall.
- 定位: D2T3
- tag: 树套树。
- 简化题意:
- 对于一个区间,快乐度定义为区间升序排序离散化后,每个数乘上序号的和。
- 计算所有区间快乐度的和。



算法 1

■ 输出 (ⁿ⁺¹₂)。

- 輸出 (ⁿ⁺¹₂)。
- 期望得分 1。

算法 2

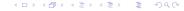
■ 暴力对于每一个区间排序离散化计算答案。

- 暴力对于每一个区间排序离散化计算答案。
- 根据实现是 $O(n^3)$ 或 $O(n^3 \log n)$, 期望得分 25, 结合 算法 1 期望得分 26。



算法 3

■ 固定左端点,右移右端点(每次加入一个数),考虑每个数 w 的序号。



- 固定左端点,右移右端点(每次加入一个数),考虑每 个数 w 的序号。
- 如果加的数 w 出现过,那就不管了。

- 固定左端点,右移右端点 (每次加入一个数), 考虑每 个数 w 的序号。
- 如果加的数 w 出现过,那就不管了。
- 否则 w 的序号是比它小的数个数 +1. 所有比 w 大的 数的序号要 +1。

- 固定左端点,右移右端点(每次加入一个数),考虑每 个数 w 的序号。
- 如果加的数 w 出现过,那就不管了。
- 否则 w 的序号是比它小的数个数 +1. 所有比 w 大的 数的序号要 +1。
- 用树状数组维护, $O(n^2 \log n)$,期望得分 45,结合算法 1 期望得分 46。



算法 4

■ 设 f_{l,r} 表示 | 到 r 这个区间的快乐度。



- 设 f_{l,r} 表示 | 到 r 这个区间的快乐度。
- $lacksymbol{\bullet}$ 令 $s_x = \sum_{i=1}^x f_{i,x}$,那么答案 $= \sum_{i=1}^n s_i$

- 设 f_{l,r} 表示 | 到 r 这个区间的快乐度。
- $lacksymbol{\bullet}$ 令 $s_x = \sum_{i=1}^x f_{i,x}$,那么答案 $= \sum_{i=1}^n s_i$
- 考虑由 *s*_{x-1} 计算 *s*_x.

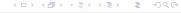
- 设 f_{l,r} 表示 | 到 r 这个区间的快乐度。
- $lacksymbol{\bullet}$ 令 $s_x = \sum_{i=1}^x f_{i,x}$,那么答案 $= \sum_{i=1}^n s_i$
- 考虑由 *s*_{x-1} 计算 *s*_x.
- 由于是一个排列,不会出现重复元素。

- 设 f_{i,r} 表示 | 到 r 这个区间的快乐度。
- \bullet 令 $s_x = \sum_{i=1}^x f_{i,x}$,那么答案 $= \sum_{i=1}^n s_i$
- 考虑由 *s*_{x-1} 计算 *s*_v.
- 由于是一个排列,不会出现重复元素。
- s_x 相比 s_{x-1} 加入一个数 w 后,首先所有区间的序号至 少为 1. (+xw). 对于所有 $A_i < w$. 会对左端点 < i 的 所有区间 w 序号 +1, (+iw), 并且对所有 $A_i > w$, 会 对左端点 $\leq i$ 的所有区间 A_i 序号 +1, $(+iA_i)$ 。

- 设 f₁, 表示 | 到 r 这个区间的快乐度。
- \bullet 令 $s_x = \sum_{i=1}^x f_{i,x}$,那么答案 $= \sum_{i=1}^n s_i$
- 考虑由 *s*_{x-1} 计算 *s*_v.
- 由于是一个排列,不会出现重复元素。
- s_x 相比 s_{x-1} 加入一个数 w 后,首先所有区间的序号至 少为 1. (+xw). 对于所有 $A_i < w$. 会对左端点 < i 的 所有区间 w 序号 +1, (+iw), 并且对所有 $A_i > w$, 会 对左端点 $\leq i$ 的所有区间 A_i 序号 +1, $(+iA_i)$ 。
- 这些用树状数组维护即可。



- 设 f₁, 表示 | 到 r 这个区间的快乐度。
- 令 $s_x = \sum_{i=1}^x f_{i,x}$, 那么答案 = $\sum_{i=1}^n s_i$
- 考虑由 *s*_{x-1} 计算 *s*_v.
- 由于是一个排列,不会出现重复元素。
- s_x 相比 s_{x-1} 加入一个数 w 后,首先所有区间的序号至 少为 1, (+xw), 对于所有 $A_i < w$, 会对左端点 < i 的 所有区间 w 序号 +1, (+iw), 并且对所有 $A_i > w$, 会 对左端点 $\leq i$ 的所有区间 A_i 序号 +1, $(+iA_i)$ 。
- 这些用树状数组维护即可。
- 时间复杂度 O(n log n), 结合算法 1, 算法 3 期望得分 60_o



算法 5

■ 考虑重复的情况怎么解决。



- 考虑重复的情况怎么解决。
- s_x 相比 s_{x-1} 加入一个数 w 后,设 i 是上一个满足 $a_i = w$ 的 i,

- 考虑重复的情况怎么解决。
- s_x 相比 s_{x-1} 加入一个数 w 后,设 i 是上一个满足 $a_i = w$ 的 i,
- 首先左端点 ≤ i 的区间显然没有影响。

- 考虑重复的情况怎么解决。
- s_x 相比 s_{x-1} 加入一个数 w 后,设 i 是上一个满足 $a_i = w$ 的 i.
- 首先左端点 < i 的区间显然没有影响。
- 只考虑左端点 i+1···x 的区间,每个元素只考虑最后 一次出现的位置的贡献(其它显然不会有贡献)

- 考虑重复的情况怎么解决。
- s_x 相比 s_{x-1} 加入一个数 w 后,设 i 是上一个满足 $a_i = w$ 的 i.
- 首先左端点 < i 的区间显然没有影响。
- 只考虑左端点 i+1···x 的区间,每个元素只考虑最后 一次出现的位置的贡献(其它显然不会有贡献)
- 然后就可以把:偏移到 0. 然后类似算法 4 的算了。

- 考虑重复的情况怎么解决。
- s_x 相比 s_{x-1} 加入一个数 w 后,设 i 是上一个满足 $a_i = w$ 的 i.
- 首先左端点 < i 的区间显然没有影响。</p>
- 只考虑左端点 i+1···x 的区间,每个元素只考虑最后 一次出现的位置的贡献(其它显然不会有贡献)
- 然后就可以把 i 偏移到 0, 然后类似算法 4 的算了。
- 但是计算需要用树套树。

- 考虑重复的情况怎么解决。
- s_x 相比 s_{x-1} 加入一个数 w 后,设 i 是上一个满足 $a_i = w$ 的 i.
- 首先左端点 < i 的区间显然没有影响。
- 只考虑左端点 i+1···x 的区间,每个元素只考虑最后 一次出现的位置的贡献(其它显然不会有贡献)
- 然后就可以把 i 偏移到 0, 然后类似算法 4 的算了。
- 但是计算需要用树套树。
- 实际上是不卡常的,如果你不小心被卡常了,还是有 85 分的。



- 考虑重复的情况怎么解决。
- s_x 相比 s_{x-1} 加入一个数 w 后,设 i 是上一个满足 $a_i = w$ 的 i.
- 首先左端点 < i 的区间显然没有影响。
- 只考虑左端点 i+1···x 的区间,每个元素只考虑最后 一次出现的位置的贡献(其它显然不会有贡献)
- 然后就可以把 i 偏移到 0, 然后类似算法 4 的算了。
- 但是计算需要用树套树。
- 实际上是不卡常的,如果你不小心被卡常了,还是有 85 分的。
- 标算是树状数组套权值线段树、但是你如果写了线段 树套线段树被卡常了,我也没办法(如果你写了线段 树套平衡树?Orz 数据结构大师) ◆□▶◆◎



前言

前言

■ tag: 分块



前言

- tag: 分块
- 定位:D1T3



■ 输出 0。



- 输出 0。
- 期望得分 1。

算法 2

■ 对于 x = 1 的数据,相当于对所有数 +z。



算法 2

■ 对于 x = 1 的数据,相当于对所有数 +z。



- 对于 x = 1 的数据,相当于对所有数 +z。
- 对于操作 1 的影响,记录所有 z 的和 s,那么询问的 答案就是 (r-l+1)s

- 对于 x=1 的数据,相当于对所有数 +z。
- 对于操作 1 的影响,记录所有 z 的和 s,那么询问的答案就是 (r-l+1)s
- 期望得分 5。



算法 3

■ 根据题意模拟。



- 根据题意模拟。
- 期望得分 20, 结合算法 2, 期望得分 25。

■ 暴力模拟会修改 $\frac{n}{x}$ 个元素,当 $x \ge 2000$,修改的元素 就不会很多。

- 暴力模拟会修改 $\frac{n}{x}$ 个元素,当 $x \ge 2000$,修改的元素 就不会很多。
- 先暴力模拟,然后前缀和解决询问。

- 暴力模拟会修改 $\frac{n}{x}$ 个元素,当 $x \ge 2000$,修改的元素 就不会很多。
- 先暴力模拟,然后前缀和解决询问。
- 期望得分 15。

算法 5

■ 考虑把计算 $\sum_{i=1}^{r} a_i$ 变为计算 $\sum_{i=1}^{r} a_i - \sum_{i=1}^{l-1} a_i$.

- 考虑把计算 $\sum_{i=1}^{r} a_i$ 变为计算 $\sum_{i=1}^{r} a_i \sum_{i=1}^{l-1} a_i$.
- 记 $s_{X,Y}$ 表示 x = X, y = Y 的 z 和, $S_{X,Y}$ 表示 $x = X, y \le Y$ 的 z 和。

- 考虑把计算 $\sum_{i=1}^{r} a_i$ 变为计算 $\sum_{i=1}^{r} a_i \sum_{i=1}^{l-1} a_i$ 。
- 记 $s_{X,Y}$ 表示 x = X, y = Y 的 z 和, $S_{X,Y}$ 表示 $x = X, y \le Y$ 的 z 和。
- 那么公差为 x 的修改对查询 $\sum_{i=1}^{n} a_i$ 的贡献就是 $S_{x,x} \cdot \lfloor \frac{n}{x} \rfloor + S_{x,n \bmod x}$ 。

- 考虑把计算 $\sum_{i=1}^{r} a_i$ 变为计算 $\sum_{i=1}^{r} a_i \sum_{i=1}^{l-1} a_i$ 。
- 记 s_{XY} 表示 x = X, y = Y 的 z 和, S_{XY} 表示 x = X, y < Y的 z 和。
- 那么公差为 \times 的修改对查询 $\sum_{i=1}^{n} a_i$ 的贡献就是 $S_{x,x} \cdot \left| \frac{n}{x} \right| + S_{x,n \bmod x}$
- 期望得分 25。

算法 6

■ 考虑结合算法 4 和算法 5。



- 考虑结合算法 4 和算法 5。
- 设置一个 s, 使得 $x \le s$ 的修改用算法 5 维护, x > s的修改用算法 4 维护。

- 考虑结合算法 4 和算法 5。
- 设置一个 s, 使得 $x \le s$ 的修改用算法 5 维护, x > s的修改用算法 4 维护。
- 但是要先考虑算法 4 怎么去掉"询问在修改之后"的 限制。

- 考虑结合算法 4 和算法 5。
- 设置一个 s,使得 x < s 的修改用算法 5 维护, x > s 的修改用算法 4 维护。
- 但是要先考虑算法 4 怎么去掉"询问在修改之后"的 限制。
- 直接用树状数组多次单点修改即可。

- 考虑结合算法 4 和算法 5。
- 设置一个 s,使得 x < s 的修改用算法 5 维护, x > s 的修改用算法 4 维护。
- 但是要先考虑算法 4 怎么去掉"询问在修改之后"的 限制。
- 直接用树状数组多次单点修改即可。
- 当 $s = \sqrt{n}$ 时,时间复杂度 $O(m\sqrt{n}\log n)$ 。



- 考虑结合算法 4 和算法 5。
- 设置一个 s, 使得 $x \le s$ 的修改用算法 5 维护, x > s 的修改用算法 4 维护。
- 但是要先考虑算法 4 怎么去掉"询问在修改之后"的 限制。
- 直接用树状数组多次单点修改即可。
- \blacksquare 当 $s = \sqrt{n}$ 时,时间复杂度 $O(m\sqrt{n}\log n)$ 。
- 在常数很优秀的情况下期望得分 100。



题解

算法 7

■ 考虑怎么优化 $x > \sqrt{n}$ 的情况的算法。



- 考虑怎么优化 $x > \sqrt{n}$ 的情况的算法。
- 这一部分修改次数是 $O(\sqrt{n})$ 每次操作的,但是修改是 O(1) 此操作。



- 考虑怎么优化 $x > \sqrt{n}$ 的情况的算法。
- 这一部分修改次数是 $O(\sqrt{n})$ 每次操作的,但是修改是 O(1) 此操作。
- 发现分块修改是 O(1) 的,询问是 $O(\sqrt{n})$ 的。

- 考虑怎么优化 $x > \sqrt{n}$ 的情况的算法。
- 这一部分修改次数是 $O(\sqrt{n})$ 每次操作的,但是修改是 O(1) 此操作。
- 发现分块修改是 O(1) 的,询问是 $O(\sqrt{n})$ 的。
- 直接用分块就是修改和询问都是 $O(\sqrt{n})$ 的了。

- 考虑怎么优化 $x > \sqrt{n}$ 的情况的算法。
- 这一部分修改次数是 $O(\sqrt{n})$ 每次操作的,但是修改是 O(1) 此操作。
- 发现分块修改是 O(1) 的,询问是 $O(\sqrt{n})$ 的。
- 直接用分块就是修改和询问都是 $O(\sqrt{n})$ 的了。
- 时间复杂度 $O(m\sqrt{n})$, 期望得分 100。



■ 本套题的题目标题参考八连测 whzzt 出的题的标题。

试题列表

赛题 #A: T13527 whzzt-Conscience | 满分: 100分

赛题 #B: T13529 whzzt-Warmth | 满分: 100分

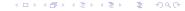
赛题 #C: T13531 whzzt-Confidence | 满分: 100分





总结

■ 本套题思维难度适中,





- 本套题思维难度适中,
- 码量适中,



The End

- 本套题思维难度适中,
- 码量适中,
- 解法自然。

- 本套题思维难度适中,
- 码量适中,
- 解法自然。
- 你可以利用这套题, 培养自己的信心。



Thanks!