数据结构题目选讲

九条可怜

杭州天水幼儿园

一些比较牛逼的均摊问题

- 有一些问题标记没有办法合并

一些比较牛逼的均摊问题

- 有一些问题标记没有办法合并
- 计算均摊复杂度

A

区间开根号（下取整），询问区间和

$n, m \le 10^5$

A

- 一个数开根号 $O(\log\log n)$ 次后一定会变成 1。

A

- 一个数开根号 $O(\log\log n)$ 次后一定会变成 1。
- 每一次找区间中所有大于 1 的数暴力开根。

A

- 一个数开根号 $O(\log\log n)$ 次后一定会变成 1。
- 每一次找区间中所有大于 1 的数暴力开根。
- $O(n \log n \log\log n)$

B

区间对 x 取模，询问区间和

$n, m \leq 10^5$

B

- 每一个数每一次取模值都至少除以 2。

B

- 每一个数每一次取模值都至少除以 2。
- 每一个数最多只会变化 $O(\log n)$ 次。

B

- 每一个数每一次取模值都至少除以 2。
- 每一个数最多只会变化 $O(\log n)$ 次。
- 每一次找区间中需要被修改的数 ($> x$ 的数) 暴力修改。

B

- 每一个数每一次取模值都至少除以 2。
- 每一个数最多只会变化 $O(\log n)$ 次。
- 每一次找区间中需要被修改的数 ($> x$ 的数) 暴力修改。
- $O(n\log^2 n)$

区间对 x 取模，区间覆盖，询问区间和

$n, m \leq 10^5$

C

- 把相同的数缩成一段。

C

- 把相同的数缩成一段。
- 每一段最多被修改 $O(\log n)$ 次。

C

- 把相同的数缩成一段。
- 每一段最多被修改 $O(\log n)$ 次。
- 每一次修改会增加 $O(1)$ 段。

C

- 把相同的数缩成一段。
- 每一段最多被修改 $O(\log n)$ 次。
- 每一次修改会增加 $O(1)$ 段。
- 平衡树维护。

C

- 把相同的数缩成一段。
- 每一段最多被修改 $O(\log n)$ 次。
- 每一次修改会增加 $O(1)$ 段。
- 平衡树维护。
- $O(n\log^2 n)$

D

区间取 $\phi(x)$，区间覆盖，询问区间和

$n, m \le 10^5$

D

- 奇数变偶数，偶数除以 2。

D

- 奇数变偶数，偶数除以 2。
- 每一段只会被修改 $O(\log n)$ 次。

D

- 奇数变偶数，偶数除以 2。
- 每一段只会被修改 $O(\log n)$ 次。
- 平衡树维护。

D

- 奇数变偶数，偶数除以 2。
- 每一段只会被修改 $O(\log n)$ 次。
- 平衡树维护。
- $O(n \log n)$。

E

区间加，区间开根号（下取整），询问区间和。

$n, m \leq 10^5$

E

- 维护最大值 ma，最小值 mi，$\mathsf{ma}-\mathsf{mi}\le 1$ 的时候可以打标记。

E

- 维护最大值 ma，最小值 mi，$\mathsf{ma} - \mathsf{mi} \le 1$ 的时候可以打标记。
- 大力 DFS。

- 维护最大值 ma，最小值 mi，ma − mi ≤ 1 的时候可以打标记。
- 大力 DFS。
- 时间复杂度 $O(n \log n \log \log n)$。

E

- 维护最大值 ma，最小值 mi，ma − mi ≤ 1 的时候可以打标记。
- 大力 DFS。
- 时间复杂度 $O(n \log n \log \log n)$。
- 复杂度证明？

E

- 考虑差分。

E

- 考虑差分。
- $|\sqrt{\lfloor a \rfloor} - \sqrt{\lfloor b \rfloor}| \leq \lfloor \sqrt{|a-b|} \rfloor$

E

- 考虑差分。
- $|\sqrt{\lfloor a\rfloor}-\sqrt{\lfloor b\rfloor}|\leq\lfloor\sqrt{|a-b|}\rfloor$
- 等号只在 $|a-b|\leq 1$ 的时候取到。

E

- 考虑差分。
- $|\sqrt{\lfloor a\rfloor} - \sqrt{\lfloor b\rfloor}| \le \lfloor\sqrt{|a-b|}\rfloor$
- 等号只在 $|a-b| \le 1$ 的时候取到。
- 差分后每一次非 1 的位置都会开根号。

F

区间 or x，区间 and x，询问区间最小值。

$n, m \leq 10^5$

F

- 如果所有数 and x 的值都相同，可以直接打标记。

F

- 如果所有数 and x 的值都相同，可以直接打标记。
- 区间所有数都相同的位 same，这些位的值 samew 以及最小值 mi

F

- 如果所有数 and x 的值都相同，可以直接打标记。
- 区间所有数都相同的位 same，这些位的值 samew 以及最小值 mi
- 如果 same and $x = x$，那么直接打标记。

F

- 如果所有数 and x 的值都相同，可以直接打标记。
- 区间所有数都相同的位 same，这些位的值 samew 以及最小值 mi
- 如果 same and $x = x$，那么直接打标记。
- 时间复杂度 $O(n \log^2 n)$。

F

- 如果所有数 and x 的值都相同，可以直接打标记。
- 区间所有数都相同的位 same，这些位的值 samew 以及最小值 mi
- 如果 same and $x = x$，那么直接打标记。
- 时间复杂度 $O(n\log^2 n)$。
- 复杂度证明？

F

- 定义势能函数 Φ 为 i 和 $i+1$ 不同的位的数量之和。

F

- 定义势能函数 Φ 为 i 和 $i+1$ 不同的位的数量之和。
- 每一次操作最多会增加 $O(\log n)$

F

- 定义势能函数 Φ 为 i 和 $i+1$ 不同的位的数量之和。
- 每一次操作最多会增加 $O(\log n)$
- 每一次线段树操作时间复杂度 $O(\log n)$ 会让 Phi 减一。

F

- 定义势能函数 Φ 为 i 和 $i+1$ 不同的位的数量之和。
- 每一次操作最多会增加 $O(\log n)$
- 每一次线段树操作时间复杂度 $O(\log n)$ 会让 Phi 减一。
- 总的线段树操作次数为 $O(n \log n)$.

二进制分组

一种在线转离线的 idea。

G

点集 S，插入点，询问与 p 的叉积最大值。

强制在线。

$n, m \leq 10^5$

G

- 最优值一定出现在凸包上.

G

- 最优值一定出现在凸包上.
- 动态凸包?

G

- 最优值一定出现在凸包上.
- 动态凸包?
- 如果有 n 个点，把 n 个点拆成长度等于 2 的幂且递减的段。

G

- 最优值一定出现在凸包上.
- 动态凸包?
- 如果有 n 个点，把 n 个点拆成长度等于 2 的幂且递减的段。
- $13 = 8 + 4 + 1$

G

- 最优值一定出现在凸包上.
- 动态凸包?
- 如果有 n 个点，把 n 个点拆成长度等于 2 的幂且递减的段。
- $13 = 8 + 4 + 1$
- 维护每一段的凸包，每插入一个数，就和 2048 一样向前合并，每一次合并每一段暴力重构凸包。

G

- 最优值一定出现在凸包上.
- 动态凸包?
- 如果有 n 个点，把 n 个点拆成长度等于 2 的幂且递减的段。
- $13 = 8 + 4 + 1$
- 维护每一段的凸包，每插入一个数，就和 2048 一样向前合并，每一次合并每一段暴力重构凸包。
- 排序可以归并，时间复杂度 $O(n \log^2 n)$.

H

点集 S，末端插入点，询问区间 $[l, r]$ 与 p 的叉积最大值。

强制在线。

$n, m \leq 10^5$

H

- 用线段树维护区间凸包。

H

- 用线段树维护区间凸包。
- 当一个区间被插满时就建这个区间的凸包。

H

- 用线段树维护区间凸包。
- 当一个区间被插满时就建这个区间的凸包。
- 时间复杂度 $O(n\log^2 n)$

点集 S，末端插入点或删除末端点，询问区间 $[l,r]$ 与 p 的叉积最大值。

强制在线。

$n, m \leq 10^5$

- 二进制分组的结构有 $O(\log n)$ 层。

- 二进制分组的结构有 $O(\log n)$ 层。
- 采取延迟重构的策略，每一层只有最后一段没有构建凸包。

- 二进制分组的结构有 $O(\log n)$ 层。
- 采取延迟重构的策略，每一层只有最后一段没有构建凸包。
- 每插慢一个区间，就重构它前一个区间。

- 二进制分组的结构有 $O(\log n)$ 层。
- 采取延迟重构的策略，每一层只有最后一段没有构建凸包。
- 每插慢一个区间，就重构它前一个区间。
- 时间复杂度 $O(n\log^2 n)$

J

加边，删边，询问两个点是否联通

$n, m \le 10^5$

K

区间对等差数列取 max，询问单点值。

$n, m \leq 10^5$

K

- 线段树维护，每一个节点维护一条直线 l，每一个节点的最大值为它往上所有区间保留的直线点值的最大值。

K

- 线段树维护，每一个节点维护一条直线 l，每一个节点的最大值为它往上所有区间保留的直线点值的最大值。
- 打标记，要比较 $l1$ 与 $l2$。

K

- 线段树维护，每一个节点维护一条直线 l，每一个节点的最大值为它往上所有区间保留的直线点值的最大值。
- 打标记，要比较 $l1$ 与 $l2$。
- 折两条直线一条控制了一个前缀，另一条控制了一个后缀，一定有一条只控制了一个孩子的一部分。

K

- 线段树维护，每一个节点维护一条直线 l，每一个节点的最大值为它往上所有区间保留的直线点值的最大值。
- 打标记，要比较 $l1$ 与 $l2$。
- 折两条直线一条控制了一个前缀，另一条控制了一个后缀，一定有一条只控制了一个孩子的一部分。
- 把这条直线 pushdown 到那个孩子就可以了。

K

- 线段树维护，每一个节点维护一条直线 l，每一个节点的最大值为它往上所有区间保留的直线点值的最大值。
- 打标记，要比较 $l1$ 与 $l2$。
- 折两条直线一条控制了一个前缀，另一条控制了一个后缀，一定有一条只控制了一个孩子的一部分。
- 把这条直线 pushdown 到那个孩子就可以了。
- $O(n\log^2 n)$

L

区间加，询问区间 gcd

$n, m \leq 10^5$

L

- $(a,b,c)=(a,b-a,c-b)$

L

- $(a, b, c) = (a, b - a, c - b)$
- 差分，变成单点修改

L

- $(a, b, c) = (a, b - a, c - b)$
- 差分，变成单点修改
- $O(n \log^2 n)$

M

矩形加，询问矩形 gcd，已知所有询问过定点 (x, y)。

$n \times m \leq 5 \times 10^5, q \leq 10^5$

M

- 以 (x, y) 为原点二阶差分整个坐标轴。

M

- 以 (x, y) 为原点二阶差分整个坐标轴。
- x 轴 y 轴维护一阶差分，(x, y) 维护实际值，四周维护二阶差分。

M

- 以 (x, y) 为原点二阶差分整个坐标轴。
- x 轴 y 轴维护一阶差分，(x, y) 维护实际值，四周维护二阶差分。
- 大分类讨论，树套树。

M

- 以 (x, y) 为原点二阶差分整个坐标轴。
- x 轴 y 轴维护一阶差分，(x, y) 维护实际值，四周维护二阶差分。
- 大分类讨论，树套树。
- $O(q \log n \log m)$

N

一个集合序列，给区间的集合都插入一个数 c，询问区间第 K 大。

$n, m \leq 10^5$

O

一棵树，路径权值翻转，询问路径和。

$n, m \leq 10^5$

O

- 用平衡树维护树链剖分。

O

- 用平衡树维护树链剖分。
- 每一次用平衡树把路径上的所有数接起来翻转再按照原来数量拆开来接回去。

O

- 用平衡树维护树链剖分。
- 每一次用平衡树把路径上的所有数接起来翻转再按照原来数量拆开来接回去。
- $O(n \log^2 n)$

P

树上 LIS。

$n \leq 10^5$

Q

区间升序排序，区间降序排序，操作结束后输出整个区间。

$n, m \leq 10^5$

Q

- 每一个有序区间用一棵权值线段树维护有哪些数。

Q

- 每一个有序区间用一棵权值线段树维护有哪些数。
- 线段树的分裂与合并。

Q

- 每一个有序区间用一棵权值线段树维护有哪些数。
- 线段树的分裂与合并。
- $O(n \log n)$

R

询问区间中每种数的出现次数的平方和。

$n, m \leq 10^5$

R

- 莫队算法。

R

- 莫队算法。
- $O(n\sqrt{n})$

S

询问 $[l,r]$ 中间有多少种不同的数字在 $[a,b]$ 之间

$n,m \leq 10^5$

S

- 莫队算法。

S

- 莫队算法。
- 分块。

S

- 莫队算法。
- 分块。
- $O(n\sqrt{n})$

- 转化成三维数点。

S

- 转化成三维数点。
- $O(n \log^2 n)$

T

n 个人去打水，第 i 个人打水需要 a_i 的时间，有 m 个水龙头。

修改某个人打水需要的时间，询问最少的总等待时间。

$n, m \leq 10^5$

T

- 排序后答案就是 $\sum_{i=1}^{n}\lfloor\frac{i-1}{m}\rfloor a_i$

T

- 排序后答案就是 $\sum_{i=1}^{n}\lfloor\frac{i-1}{m}\rfloor a_i$
- 块状链表，根据 m 和 $\sqrt{n}$ 的大小关系分情况讨论。

T

- 排序后答案就是 $\sum_{i=1}^{n}\lfloor\frac{i-1}{m}\rfloor a_i$
- 块状链表，根据 m 和 $\sqrt{n}$ 的大小关系分情况讨论。
- $O(n\sqrt{n})$

谢谢大家