数据结构选讲

线性基 简洁的代码实现与本质

前后缀线性基 Ivan and Burgers code : from_yixiuge777

P5607 更优秀的做法_chenxinyang2006

平衡树

scapegoat 优雅的暴力

treap 随机的艺术

splay 旋转

fhq_treap 分裂与合并

《浅谈Splay与Treap的性质及其应用》 董炜隽 国家集训队2018论文

Finger Search 时间其实与树上两个特定节点距离有关 则 splay 和 treap 合并时间复杂度为 $O(n \log n)$ 。

平衡树维护序列翻转

线段树

维护的信息的特点

线段树维护分治信息略解 铃悬

《范围修改查询问题》 李欣隆

楼房重建 二次递归更新

《区间最值操作与历史最值问题》 吉如一 国家集训队2016论文

线段树优化建图 线段树合并 动态开点线段树

分块 平衡复杂度 莫队

扫描线方向

区间加区间 sin 和

序列。维护区间加，区间 $\sum_{i=l}^{r} \sin(a_i)$。$n \leq 200000$。

为了让信息封闭能够合并还需要维护 cos 和。

code : from_yixiuge777

幻梦 | Dream with Dynamic

序列。维护区间加，区间每个数变成自己的 popcount ，单点询问。$n \leq 300000, q \leq 1000000, a_i, x \leq 10^9$。

popcount 操作的特点是进行之后值域变成 $O(\log v)$。则在线段树上维护加法标记和 popcount 标记，第一次 popcount 操作之后值域变成 $O(\log v)$，维护这些值经过之后操作的答案的函数。函数复合封闭，标记可以下传。时间复杂度 $O(n \log n \log v)$。

code : from_yixiuge777

统一省选（bad）2023sd一轮省集d3t2

游戏中有一个人，其具有生命值 H。初始时，H 的值为 H_0。游戏包含 n 个关卡，每个关卡可能有三种类型。

• 1 a_i：这是一个难度为 a_i 的 1 类关卡。如果你当前的生命值 H 小于等于 a_i，则你会死亡。否则，你将失去 a_i 点生命值。形式化地，如果 $H \leq a_i$，那么你死亡。否则 $H := H - a_i$。

• 2 a_i：这是一个难度为 a_i 的 2 类关卡。如果你当前的生命值 H 小于 a_i，则你会死亡。否则，你的生命值将变为 a_i。形式化地，如果 $H < a_i$，那么你死亡。否则 $H := a_i$。

• 3 a_i：这是一个难度为 a_i 的 3 类关卡。由于发生了不可描述的事件，你很有精神，因此如果你的生命值 H 不足 a_i，则你的生命值将变为 a_i。形式化地，$H := \max(H, a_i)$。

给你 n 个关卡的信息，你要支持 q 次以下操作：

• 1 $x_i t_i a_i$：修改第 x_i 个关卡的类型为 t_i，难度为 a_i。

• 2 l_i：现在假设关卡 $1, 2, ..., l_{i-1}$ 全部消失，从关卡 l_i 进行一次游戏，并按照顺序依次进行关卡 $l_i, l_{i+1}, ..., n$。你想要知道最大的 r，使得在第 r 个关卡结束时你还活着。特别地，如果你在完成了关卡 l_i 后便死亡，则输出一行 -1。

$n, q \leq 10^6, H, a_i \leq 10^{12}$。

发现三种函数的复合封闭，都可以写成如下形式：

$$f(x)=\begin{cases}-\infty & x\le a\\ y & a<x\le b\\ x-b+y & b<x\end{cases}$$

信息可合并。线段树维护。时间复杂度 $O(n\log n)$。

code : from_yixiuge777

[Ynoi2013] 对数据结构的爱

以下错误程序段：

```
Function ModAdd(x,y,p)
    if x+y<p return x+y
    return x+y-p
Function Sum(A,l,r,p)
    ans=0
    for i from l to r do
        ans=ModAdd(ans,a[i],p)
    return ans
```

当 x,y 过大时 ModAdd 函数会出现错误。多次询问以上错误程序 Sum(A,l,r,p) 的返回值。$n\le 1000000, m\le 200000, p\le 10^9$。

只需要计算 $-p$ 的次数。对于一段长为 len 的区间，只有 len 种结果。并且初始输入数字越大，次数越多。维护每个 $-p$ 次数对应的最小输入值。这个信息满足半群性质可以合并。时间复杂度 $O(n\log n+m\log^2 n)$。

code : from_yixiuge777

「C.E.L.U-02」苦涩

维护 n 个初始为空的可重集。

- $[l,r]$ 的可重集都加入 k 数字。
- $[l,r]$ 中的每个可重集，如果其最大值等于区间可重集最大值，删去该集合内一个最大值，有多个只删除一个。
- 查询 $[l,r]$ 中数字最大值。

$n,m\le 200000$。

用堆维护最大值。有区间操作，线段树，每一个结点都是一个大根堆。插入的时候找到插入区间分成的线段树上对应的小区间，插入到对应的堆里。但是注意这里的标记很难下传，使用标记永久化，表示子树里的所有结点实际上都有这个值，询问的时候只需要把路径上所有的最大值都取最大值即可。

麻烦的是删除操作。先找到这一段的最大值，暴力的想法是依次移除每个标记，但是一个区间的最大值小于要删除的值的时候就不操作了。

这样就是对的了。首先线段树上的元素个数是线性对数级别的，只要删除操作能够快速的找到应该被删除的元素在哪里，时间就是对的。如果整体的最大值比要删除的值小了就推出，这样找到每个要被删除的元素的时间是对数级别的，所以均摊删除操作是线性对数平方级别的。

但是对于一个区间可能会对其部分删除，多删的一部分还需要加回来。只有在区间两端的两个线段树节点需要进行这样的操作，增加的元素数量是线性对数级别的。总时间复杂度 $O((n+m)\log^2 n)$ 。

code : from_yixiuge777

[CERC2017]Intrinsic Interval

一个排列上的好区间为将区间内的数排序之后其为连续正整数。多次询问一个包含一个区间的最短好区间。$n \leq 100000$。

把值域连续的区间叫做好区间。关键性质：两个好区间的交也是好区间。因为序列上是两个区间，值域上也是两个区间，交起来也是好区间。

对于一个询问 $[l,r]$ ，一个暴力想法是从 r 开始往大找所有能包含的候选答案区间，选一个最小的。而第一个找到的就是最小的！如果第一个找到的区间是 $[L,R]$ ，后面又找到了一个更短的区间 $[L',R']$ ，其中 $R'>R, L'>L$ ，那么区间 $[L',R]$ 是两个区间的交也一定是一个好区间，而且一定非空。这与从 R 处找到的最小的包含的区间 $[L,R]$ 矛盾。所以第一个找到的 $[L,R]$ 就是答案。换句话说，R 最小的那个就是最短的好区间。

看另外一个问题：如何快速判断好区间？值域连续太难以维护。因为是排列，值域连续等价于：有 $r-l$ 个二元组 $(v,v+1)$ 。于是我们只需要维护 $r-l-$二元组的数量，看这个值是否为0，即可判断这是不是一个好区间。

离线扫描线右端点，询问按右端点排序。线段树维护 $[l,i]$ 的上述的值，同时维护区间最小值和区间最小值出现的最右的位置。每次扩展一个右端点，更新线段树，然后将所有 $r \leq i$ 的询问拿出来，按 l 从大到小依次寻找对应的答案。如果一个询问找到了答案，这就是最终答案。如果没找到，那么 l 更小的询问也一定找不到。用线段树和优先队列维护。时间复杂度 $O(n\log n)$ 。

code : from_yixiuge777

[THUPC2022 决赛] rsraogps

给序列 $a_1,\ldots,a_n$，$b_1,\ldots,b_n$，$c_1,\ldots,c_n$，定义区间 $[l,r]$ 的价值为 $a_l,\ldots,a_r$ 按位与，$b_l,\ldots,b_r$ 按位或，$c_l,\ldots,c_r$ 的最大公因数，这三者的乘积；m 次查询，每次查询给出区间 $[l,r]$，查询满足 $l\le l'\le r'\le r$ 的 $[l',r']$ 的价值之和。$n\le 1000000, m\le 5000000$。

右端点从左到右扫描线。对于每个位置维护 s_i 表示 $l\le i$ 的所有 $[l,r]$ 的答案和。扫描线扫描到询问的时候答案即为 s_r-s_{l-1}。考虑右端点向后移动一位 s_i 的变化情况。此时有部分后缀的区间答案发生改变，由位运算和最大公约数的性质这样的改变只有 $O(n\log v)$。暴力处理这些改变。对于前面没有改变的位置，其 s_i 应加上一个值，值为 $l\le i$，r 为右端点的区间的答案和。则其可以表示为 $s_i+k_i\cdot(T-last_i)$，$last_i$ 表示上次修改的时间。时间复杂度 $O(n\log v+m)$。

我去年写的题解：

离线扫描线 r。每次只会有最多 $\log n$ 个 $[l,r]$ 的权值会变。维护 s_i 表示 l 的前缀和，为所有 $l\le i$，所有已经处理过的 r 的组成的所有区间的和，那么一个询问的答案即在扫描到 r 处时 s_r-s_{l-1} 的值。注意这里 s_i 的定义类似于二维前缀和。

观察如何维护 s_i 的值。每次扩展 r 只会有靠近 r 的几个 l 对应的 $[l,r]$ 的权值变化，前面的大多数 $[l,r]$ 的权值都没有变化，那么可以维护 pre_i 表示当前 r 时，$l\le i,[l,r]$ 的权值和。注意这里 pre_i 是一维的前缀和。如果 i 位置的答案不变，前面的答案也不会变，每次 s_i 增加的量就一直是 pre_i。将 s_i 比作一次函数，$s_i=pre_i*(T-lat_i)+B_i$，其中 lat_i 为上次更新此处答案的时间。由此每次扩展 r 只需要 $O(1)$ 更改 $O(\log v)$ 个 l 位置的值就可以了。查询 $O(1)$。总时间复杂度 $O(n\log v+m)$。

code : from_yixiuge777

[HEOI2016/TJOI2016] 排序

排列。进行若干次局部排序，排序一个区间内的数升序或降序排列。最终问 a_{pos} 的值。$n,m\le 100000$。

二分答案。变成查询区间和和区间覆盖操作。时间复杂度 $O(n\log^2 n)$。

code : from_yixiuge777

「C.E.L.U-02」划分可重集

长度为 n 的数列 $\{v\}$ 划分成两个可重集 a 和 b，每个数必须至少被划分进一个可重集中。

一个数 v_i 可以被划分进 a 当且仅当 $j<i\ and\ v_j\le v_i-k$ 的 v_j 都没有被划分进 a。一个数 v_i 可以被划分进 b 当且仅当 $j<i\ and\ v_j\ge v_i+k$ 的 v_j 都没有被划分进 b。同时给出了 m 组关系，每组关系代表 u 和 v 不能划分进同一个可重集里。求能使划分成功的最小的 k 或无解。$n\le 20000$。

二分答案。优化建图 + 2-sat 。

对于二维偏序形式，可以使用 cdq 分治每次前缀后缀优化跨过分治中心的边，也可以使用可持久化线段树优化建图。时间复杂度 $O(n \log n \log v)$。

code : from_yixiuge777 使用可持久化线段树实现。

[Ynoi2019 模拟赛] Yuno loves sqrt technology I

排列静态在线区间逆序对。$n \leq 100000$。

序列分块。预处理所有块边界之间的答案。预处理每个数在块内与前缀和后缀的贡献。预处理前 i 个数与第 j 块的的数的贡献。

询问时，若两数在同一块内，对每个数用预处理的块内前缀答案累加，减去多算的区间与块左端点到区间左端点的答案。后者可以归并排序。若两数不在同一块内，中间整块已经预处理好了答案，散块之间可以使用预处理的块内前缀后缀贡献，散块对整块使用前 i 个数与第 j 块的的数的贡献。散块归并排序。每个块预处理排序好的数组省去排序。时间复杂度 $O(n\sqrt{n})$。

code : from_yixiuge777

[Ynoi2019 模拟赛] Yuno loves sqrt technology III

序列静态在线区间众数。$n \leq 500000$。

预处理块间答案。对于散块数，每出现一个最多使答案加一。维护相同数字的出现位置，寻找散块数相同的后 ans 个数在不在区间内，在则更新答案。时间复杂度 $O(n\sqrt{n})$。

code : from_yixiuge777

Jumping Through the Array

序列 a_i 和排列 p_i。

- 求 $\sum_{i=l}^{r} a_i$。
- 在 $i->p_i$ 组成的图上，给所有 u 能到达的点的 a_x 加 C。
- $\mathrm{swap}(p_i, p_j)$。

$n \leq 200000$。

操作时间分块。

对于操作2和3的点组成关键点，从一个关键点的下一个点开始，到再一个关键点结束，所有的点缩起来，在一块操作中这些点的修改等价，操作3可以直接改，操作2在块上整体修改。如果一个环上没有关键点，这么多操作与这个环没关系，直接不用管。对于操作1，首先预处理没有处理这些操作前的前缀和，然后找每个块在这个区间的出现次数乘上这个块的加的值。所有操作完了之后暴力每个点修改其权

值，并且重新处理前缀和。唯一的问题是怎么找一个块里在一个区间中出现的数的个数。每个询问查分，变成前缀，这样的区间只有根号。对每个点加到其块的贡献上，前缀和。

总时间复杂度 $O(n\sqrt{n})$ 。

code : from_yixiuge777

[省选联考 2023] 人员调度

树上每个节点有一个集合的数。可以将一个集合中的一个数下放到子树中某点的集合中。树的权值是所有集合中数最大值的和，如果某一集合空则无贡献。每次更改会加入一个数到某点的集合中，或是删除一个数。求每次更改后经过若干次下放操作后树的最大权值。下放操作不会实际改变树上数的位置。$q,n\leq 100000$。

尝试静态问题如何处理。对于叶节点保留最大值，对于内部节点，将其集合中多的数尽可能下放到子树中。如果子树中全满了，则选择其中小的一些数扔掉腾出位置。换句话说，每个点维护了一个子树贡献答案的集合，从下往上合并，每次答案集合不能超过子树大小。可并堆维护，时间复杂度 $O(qn\log n)$ 。

如果只有加点如何处理。如果其到根的路径上所有点的答案集合都没有顶到子树上界，则这个数可以被顺利插入到到根的所有点的答案集合内。否则，遇到第一个到根路径上答案集合顶到子树大小上界的节点，其需要与集合内最小值比较大小。如果更小则无影响，如果比其大则成功替换掉。并且因为该点在答案集合内，其在之后的道路上也一定没有被淘汰，不需再考虑后续过程。

线段树分治变删除为撤销。需要支持链加链 min ,查询子树最小值。树剖加线段树实现。时间复杂度 $O(n\log^3 n)$。

code : from_yixiuge777