分块 & 分治选讲

huhao

August 8, 2022

前言

~~这次讲课范围和你们前面上的那节“数据结构的应用”有大量重叠，我重点就放在分块上。~~

本次讲课内容：

序列分治 $\times$

树分治 $\surd$

序列分块 $\surd$

树分块 $\times$

本次讲课以习题为主。

分治

分治

分治内容较为基础，不细讲，可以针对下面（或是其它没有写出来的）不懂的提问：

- cdq 分治
- 二进制分组
- 树分治
- WQS 二分
- 二分有理数
    - Stern–Brocot tree

分块

带修改序列分块

分块可以理解为一个三层，根度数为 w，第二层度数为 $\frac{n}{w}$ 的线段树。

续

为什么用分块而不用复杂度更加优秀的分治结构？

- 平衡时间复杂度。
- 便于维护信息。
  - 在修改时只需要支持 $\frac{n}{w}$ 个信息的合并，合并时支持 $\frac{n}{w}$ 个信息的合并。
  - 信息一共只需要经过两次处理，不需要要求信息必须可以多次相加。

例

单点修改查询前缀和，序列长度 10^8，修改查询次数分别为：$10^8/10^4, 10^6/10^6, 10^4/10^8$。

如果区间修改呢？

非带修序列分块

通过预处理加速询问。

同样可以考虑一下优势区间：不方便合并大量元素产生的信息，或是存储信息花费较高等等。

因为基于分治有一个特别优秀的做法：在线段树上每一个节点记录过中点的前/后缀和，就合并得到所有过中点的答案了。这种做法的缺点很明显：最复杂的合并是最后进行的。

莫队

由于分块中散块的处理需要支持多次加入单个元素，所以分块的功能会被莫队完全包含。

但是莫队只能处理离线内容。

类似莫队的分块

如果某道题离线情况下可以被莫队处理，那么可以每 $n^{\frac{2}{3}}$ 个元素分为一块，然后处理出以所有块端点作为左右端点时，莫队算法所需要的信息。

对于一个 01 序列，令 $n = 2^w$ 每 $\frac{w}{k}$ 个元素分为一块，一共有 $\dfrac{2^w k}{w}$ 块。其中一共有 $\sqrt[k]{n}$ 种不同的块。

可以尝试通过预处理每一个不同的块的信息达到加速的目的。

例

有以下操作 q 次:

- + x y z a b: 新增一只怪物。如果这是第一只新增的怪物，那么它的编号为 1；否则它的编号为最后一只新增的怪物的编号 +1。这只怪物位于魔塔的第 x 层，它的等级为 y 级，它的难度为 z。如果玩家选择击杀这只怪物，那么需要消耗 a 点血量，在击杀成功后，玩家将得到一支可以恢复 b 点血量的药剂并立即使用。
- - k: 删除编号为 k 的怪物。
- ? g l d: 表示一个询问。某玩家希望击杀魔塔前 g 层中所有等级不超过 l 且难度不超过 d 的怪物。玩家可以按照任意顺序去击杀这些怪物，登上新的一层不需要杀光当前层的所有怪物，且作战过程中不会受到别的怪物的干扰。你的任务是帮助该玩家计算出征前勇士的血量最少是多少。如果某个时刻勇士的血量是负数，那么游戏结束，你一定要防止这种情况的发生。

请写一个程序，依次回答每个询问。注意：每个询问只是玩家的一个思考，不会真正击杀任何一只怪物。

数据范围： $q \leq 1.5^5, x, y, z \leq 10^4$。

操作分块

在预处理后，如果一个操作对已经处理完的结果影响不大，那么可以尝试对操作进行分块，每一个块开始时进行初始化。

例

一颗带权有根树，支持两种操作：

1. 改变某个点的父亲，或是修改某个权值。
2. 求某个点到它 k 级祖先中，权值大于 x 的元素个数。

要求 $O(n^{\frac{5}{3}})$。

解

每 $n^{\frac{1}{3}}$ 个操作分为一块，预处理出每一个点的 $n^{\frac{1}{3}}$ 级祖先，每次修改就标记一下它的 $0 \sim n^{\frac{1}{3}}$ 级祖先，判断能否到祖先路径是否被修改就查询一下祖先是否有标记即可。

考虑一个点的路径，显然标记只会有 $n^{\frac{1}{3}}$ 段，所有会有 $n^{\frac{1}{3}}$ 次连续 $n^{\frac{1}{3}}$ 步走向父亲，以及 $\frac{n}{n^{\frac{1}{3}}} = n^{\frac{2}{3}}$ 次走向 $n^{\frac{1}{3}}$ 级祖先的操作。

总复杂度 $O(n^{\frac{5}{3}})$。

tricks

根号分治

对于取值为 $[1,n]$ 的变量 x，当 $x < w$ 和 $x \geq w$ 时分别采用两种方法。

例

你一开始在 $(0,0)$，你可以移动 n 次，每次移动 $(1,1)$ 或 $(1,-1)$，且不能到达 $(k,-1),(k,m)$，对于每一个 i，求到达 (n,i) 的方案数。

例

一个序列，需要支持区间求和，和区间每隔 k 个元素加同一个数。

分散层叠

对于 n 个有序序列 a_i，可以在 $O(\sum|a_i|)$ 的预处理后 $O(n+\log\max|a_i|)$ 的时间复杂度内求出每一个数组中大于给定值的元素个数：

令 b 为由以下方式生成的序列：b_i 由 a_i 以及 b_{i-1} 偶数位归并构成（或是每 k 个取 1 个）。

在查询 x 时，先查询 x 在 b_n 中的排位，然后就可以 $O(1)$ 得到 b_{n-1} 中的排位，以此类推。

例

区间加，求区间第 k 小。

解

分块，分为 w 块，然后建立分散层叠的 b 数组。

每一次修改，都会出现新的两个块，并对若干个块整体加一个数。

新的块直接处理分散层叠。若干个块整体加一个元素，可以在末尾的分散层叠数组中所有元素都加上一个值，这样就可以快速还原了。

每一次修改都会给分散层叠的数组数加 3，不妨考虑每 w 次修改重构一次，这样均摊单次复杂度：

$$O(\frac{1}{w}n + \frac{n}{w} + \log n(w + \log n)) = O(\frac{n}{w} + w\log n)$$

当 $w = \sqrt{\frac{n}{\log n}}$ 时，均摊单次复杂度为 $O(\sqrt{n\log n})$

例题

分糖果

两个序列 a, c，若干次操作，每一次操作会给定一个区间 $[l, r]$ 和一个数 x，对 $i \in [l, r]$ 使 a_i 加 x，并让它在 $[0, c_i]$ 中（大于 c_i 变成 c_i，小于 0 变成 0）。

求最后的 a 数组。

解

依次考虑 a 的每一个元素，用一个数据结构维护一下操作。

这里维护可以考虑一下操作序列 $b_1 \dots b_m$，若它的一个子区间 $[l,r]$ 和大于等于 c_i（或小于等于 $-c_i$），那么在经过 r 次操作后一定是 c_i（或 0）。

然后考虑维护一下第一次取到 $0/c_i$，在考虑最后一次取 $0/c_i$ 的时候即可。

开店

给定一颗带权树，边有长度，多次询问：

所有权值在 $[l, r]$ 中的点，和 x 的距离和。

强制在线，$n \leq 1.5 \times 10^5, q \leq 2 \times 10^5$。

解

考虑动态给点加权，动态询问权距离和，这就是一个经典的点分树问题。

给点分树套用可持久化即可。

农民

小 D 在家种了一棵二叉树，第 i 个结点的权值为 a_i。

小 D 为自己种的树买了肥料，每天给树施肥。

二叉搜索树专用版肥料是这么工作的：首先，假设所有节点权值互不相同（小 D 的二叉树可能不满足），每种权值对应一种肥料，所有肥料会从根进入树中，如果一种肥料对应的权值等于当前结点权值，这种肥料会被当前结点完全吸收，否则若肥料对应的权值小于当前结点权值，肥料会流向左子树，否则流向右子树，如果流向的子树为空，肥料只好流失蒸发了。显然，如果树是二叉搜索树，所有节点都能吸收到肥料。

小 D 觉得自己还能抢救一下，他会进行若干次操作，每次操作修改一个点的权值或者翻转一个子树（子树内所有节点左右儿子互换）。在操作过程中，他有时会想知道一个点当前是否能吸收到肥料，以决定之后如何操作，请你帮帮可怜的小 D 吧。

$n, q \leq 10^5$。

解

树链剖分，记录一下从链首到链尾每一个节点肥料的上下界，以及反转后的值。

反转标记可以类似 lct 地维护：访问时下传。

幻想乡战略游戏

给定一颗带权树，边有长度，多次询问：

如果给第 u 号点权值加 x，那么这个图的带权重心（距离所有点权距离和最近）是哪个点。

每个点度数小于等于 20，$n, q \leq 10^5$。可以考虑一下没有点度数小于 20 的条件。

解

在点分树上向点权和大于一半的方向移动即可。

不难发现，可以将一个点拆成两个相连且边权为 0 的点重心不变，这样可以不断将一个点度数减小。

回转寿司

你有一个序列 a，每一次会给出一个区间 $[l, r]$ 并给出 A，你需要：

递增的对每一个 $i \in [l, r]$ 判断：若 $a_i > A$ 则交换 a_i, A。

你需要输出上述步骤后的 A。

$n \leq 4 \times 10^5, Q \leq 25000$，9s。

解

分块，对每一块分别考虑下面两个新的问题：求操作的答案，求序列。显然后者的次数远远小于前者。

首先，具体的答案肯定是没有问题的：考虑初始的 A 被加入 a，最终的 A 被剔除 a。

其次，可以考虑记录所有的操作，假设它们形成操作序列 $A_1 \ldots A_m$，并依次考虑块中的每一个元素 a_i：不难发现，这是原来的问题！使用上面的办法即可做到 $O(q\sqrt{n}\log n)$。

Innophone

有一个二元函数 $f(x,y)$，它是这么定义的：

$$f(x,y)=\begin{cases} a, & \text{if} & a\le x \\ b, & \text{else if} & b\le y \\ 0, & \text{else} & \end{cases}$$

其中 a,b 为常数。现在给定 n 组 x,y，你需要选择合适的 a,b，使得 $\sum_{i=1}^{n} f(x_i,y_i)$ 最大。

$n\le 1.5\times 10^5$。

解

这个问题可以等效为:

一个初始全 0 的序列 v，每次给 $x \le a$ 地所有 v_x 加上 x，并查询 v 中的最大值。

分块可以简单维护。

五彩斑斓的世界

一个序列，你需要：

1. 区间将大于等于 x 的元素减少 x。
2. 区间查询 x 出现次数。

值域与序列长度同阶。

要求 $O(n^{1.5})$

解

分块，考虑每一块的最大值 m，在经过一次 1 操作后，m 不大于 $\max(m-x,x)$。

所以只要维护依次询问的代价是 $O(\min(m-x,x))$ 的，那么无论进行多少次操作，总操作代价就是 $O(m)$ 的。

对 $x<m-x$ 和 $x\geq m-x$ 分别考虑即可。

这是我自己的发明

一颗带权树，支持若干次操作：

1. 换根。
2. 查询两颗子树内，分别选出两个权值相同的点的方案数。

要求 $O(n^{1.5})$。

解

显然，这个问题可以变为查询两个前缀权值相同的点的方案数。

莫队应该是最简单的做法。

未来日记

一个序列，你需要：

1. 区间将等于 x 的元素变为 y。
2. 区间查询第 k 小。

要求 $O(n^{1.5})$。

解

给序列和值域都分块，维护：

1. 每一个值在前若干序列块内出现次数。
2. 每一个值的块在前若干序列块内出现次数。

这样修改是对值域中 $O(1)$ 乘上序列中 $O(\sqrt{n})$，询问是值域中 $O(\sqrt{n})$ 乘上序列中 $O(1)$。

不归之人与望眼欲穿的人们

一个序列，你需要：

1. 单点修改。
2. 查询 or 值大于 x 的最短区间长度。

$n, q \le 5 \times 10^4, a \le 2^{30}$。

解

一个常识，一个端点固定时，不同的 or 值种类数只有 30（值域的 $\log$，下记作 $\log a$）种。

分块，如果区间在一个块内，可以通过预处理的方式求解。否则，一定过块的某一个端点，枚举那个端点向左延申多少，即可做到单次 $O(w \log a)$，其中 w 是块数。

修改时重构整块，然后维护出某一块的右端点向右的 $\log a$ 个 or 值不同的区间。

天降之物

一个序列，你需要：

1. 将所有 x 改为 y。
2. 查询最靠近的 x 和 y 的距离。

$n, q \leq 10^5$。强制在线。

解

如果询问的 x, y 出现次数均小于 $\sqrt{n}$ 次，将它们出现位置合并即可。

否则可以 x, y 有一个出现次数大于 $\sqrt{n}$ 次，可以预处理出所有这种情况的答案。

考虑修改中合并的情况：

如果 x, y 出现次数均不大于 $\sqrt{n}$，或均大于 $\sqrt{n}$，暴力修改即可。

否则将它们均保留下来，查询时分别查询，合并时就合并在出现次数不大于 $\sqrt{n}$ 的上面，如果合并后大于 $\sqrt{n}$ 就全部合并起来。

D2T2

你有一个序列，有若干询问：

在区间 $[l, r]$ 中，仅保留 $[L, R]$ 的数，最大区间子段和是多少。

$n \leq 10^5$。

解

考虑分成 $\sqrt{n} \times \sqrt{n}$ 的块，每一块都只有 $\sqrt{n}^2$ 种不同的 $[L, R]$ 的答案。

维护一段的信息可以分治求解：$T(n) = 2\,T(\frac{n}{2}) + O(n^2) = O(n^2)$（考虑 $\sum_i 2^i(\dfrac{n}{2^i})^2$）。