

数据结构 (2)

长郡中学 周书予

2019 年 9 月 9 日



CDQ 分治

CDQ 分治的基本思想是考虑左侧对右侧的贡献/影响。

常见的使用方式是处理偏序问题，例如三维偏序可以通过在外部对第一维排序，第二维内部归并，第三维维护序列数据结构的方式解决。

bzoj4237 稻草人

description

给平面上 n 个点 (x_i, y_i) , 求有多少个矩形满足其左下角和右上角都有一个点, 且矩形内部不包含别的点。

constriction

$1 \leq n \leq 2 \times 10^5, 0 \leq x_i, y_i \leq 10^9, x_i, y_i$ 互不相同。

bzoj4237 稻草人

description

给平面上 n 个点 (x_i, y_i) , 求有多少个矩形满足其左下角和右上角都有一个点, 且矩形内部不包含别的点。

constriction

$1 \leq n \leq 2 \times 10^5, 0 \leq x_i, y_i \leq 10^9, x_i, y_i$ 互不相同。

solution

按 x 坐标分治, 每层分治考虑所有跨越分治分界线的矩形。

处理某层分治时, 先把所有点按 y 坐标排序 (可以自下而上归并)。对左右两侧分别维护一个单调栈, 枚举右侧点作为右上角的点, 可以发现可与之形成合法矩形的是左侧单调栈的一段后缀, 二分即可。

整体二分

所有询问连同修改一起二分答案，每次对于答案落在 $[l, r]$ 区间内的所有询问，计算其答案是会落在 $[l, mid]$ 还是 $[mid + 1, r]$ ，同时要记录 $[l, mid]$ 中的修改对 $[mid + 1, r]$ 中的询问的影响。

注意整体二分单次 check 的复杂度只能与当前区间长度有关。以及，在分治的全过程中， mid 指针的总移动量是 $O(n \log n)$ 的。

动态区间 kth

description

维护一个序列 $\{a_i\}$ ，支持单点修改以及查询区间内第 k 大值。不强制在线。

constriction

$1 \leq n, m \leq 10^5, 0 \leq a_i \leq 10^9$.

动态区间 kth

description

维护一个序列 $\{a_i\}$ ，支持单点修改以及查询区间内第 k 大值。不强制在线。

constriction

$1 \leq n, m \leq 10^5, 0 \leq a_i \leq 10^9$.

solution

从 $[0, 10^9]$ 开始分治，当分治到 $[l, r]$ 时，拿树状数组统计每个询问区间里有多少个数 $\geq mid = \lfloor \frac{l+r}{2} \rfloor$ ，从而可以知道每个询问的答案是 $\geq mid$ 还是 $< mid$ ，向两边递归即可。

启发式合并/最大值分治

合并两个集合时，暴力枚举较小集合中的每个元素加入较大集合。这样每当一个元素被暴力枚举一次后它所在集合大小就会扩大一倍，因而暴力枚举的总个数为 $O(n \log n)$ 。

同样的，在某些分治的过程中，取 $mid = \lfloor \frac{l+r}{2} \rfloor$ 不一定最合适，而选取其他位置（比如序列的最大值）由不能保证复杂度。但是如果每层分治的复杂度只与较小的一侧的大小有关，那么这个复杂度就等同于启发式合并的复杂度。

luogu4755 Beautiful Pair

description

给一个序列 $\{a_i\}$, 求有多少对 $1 \leq i \leq j \leq n$ 满足 $a_i a_j \leq \max_{k=i}^j a_k$ 。

constriction

$1 \leq n \leq 10^5, 1 \leq a_i \leq 10^9$.

luogu4755 Beautiful Pair

description

给一个序列 $\{a_i\}$, 求有多少对 $1 \leq i \leq j \leq n$ 满足 $a_i a_j \leq \max_{k=i}^j a_k$ 。

constriction

$1 \leq n \leq 10^5, 1 \leq a_i \leq 10^9$.

solution

枚举最大值, 考虑所有跨过最大值的 (i, j) 对, 枚举较小的一侧, 另一侧的数需要在某段值域范围内, 使用可持久化线段树维护即可。

线段树分治

又称按时间分治，即，把每次询问看做一个时间点，每次修改对应一段询问的时间区间。这样一次修改就会对应到线段树上 $O(\log n)$ 个节点上，需要实现一个对节点“打标记”的过程。

现在每个节点上有了若干的修改标记，有两种处理方式：

- 如果每次修改对询问的贡献独立，那么可以在每个节点上维护某种结构，最后对于一次询问，只需要查询从线段树根到它对应的叶子这条路径上 $O(\log n)$ 个节点上的结构即可。
- 如果修改支持快速单点增量，那么可以在线段树上 dfs 并维护某种结构，进入叶子节点后即可查询询问答案，回溯时需要栈序撤销修改。

ctsc2016 时空旅行

description

你需要维护 n 个集合，集合元素为二元组 (x, v) 。集合 i 的产生方式是以某个原有集合 p_i 为样本，扩展或删除一个元素后得到新集合。有 q 次询问，每次给出 y 并指定一个集合 i ，要求从集合 i 中找出一个元素，最小化 $(x - y)^2 + v$ 。

constriction

$1 \leq n, q \leq 10^6$.

ctsc2016 时空旅行

description

你需要维护 n 个集合，集合元素为二元组 (x, v) 。集合 i 的产生方式是以某个原有集合 p_i 为样本，扩展或删除一个元素后得到新集合。有 q 次询问，每次给出 y 并指定一个集合 i ，要求从集合 i 中找出一个元素，最小化 $(x - y)^2 + v$ 。

constriction

$1 \leq n, q \leq 10^6$.

solution

集合的生成方式可以抽象成一棵树，对于某一个元素 (x, v) ，它在树上的出现位置是某一棵大树内删去若干小子树，可以发现所有元素在树上的出现区间的总个数是 $O(n)$ 的。把元素按照 x 排序后插入线段树，在每个线段树节点上维护凸壳，再在外部对所有询问按 y 排序，可以避免每次查凸壳时的二分。

codeforces938G Shortest Path Queries

description

有一张 n 点 m 条边无向连通图，边带边权，要求支持 q 次如下三种操作：（保证任何时刻图连通且无重边自环）

- 加入一条连接 x, y 边权为 z 的边。
- 删除连接 x, y 的边。
- 询问 x 到 y 的异或最短路。

constriction

$$1 \leq n, m, q \leq 2 \times 10^5, 0 \leq z < 2^{30}.$$

codeforces938G Shortest Path Queries

description

有一张 n 点 m 条边无向连通图，边带边权，要求支持 q 次如下三种操作：（保证任何时刻图连通且无重边自环）

- 加入一条连接 x, y 边权为 z 的边。
- 删除连接 x, y 的边。
- 询问 x 到 y 的异或最短路。

constriction

$1 \leq n, m, q \leq 2 \times 10^5, 0 \leq z < 2^{30}$.

solution

两点的路径异或和是这两点的某一条路径再异或上若干个环的异或和。

对每条边找出其存在的区间并放到线段树上，再对整棵线段树做一遍 dfs，每 dfs 到一个节点就加入这个节点上的边，这样就可以把删边操作变为栈序撤销。维护带权并查集和一个线性基即可。

二进制分组

可以用于维护一些合并较慢的信息（比如凸包）。

每新插入一个元素后，先令其自成一组，然后从小到大不断合并大小相同的两组。

可以使用线段树来维护这个结构，这样同时也可以记录下所有曾出现过的组。

uoj46 玄学

description

有一个序列 $\{a_i\}$, 支持 q 次如下两种操作:

- 加入一个变换: 将区间 $[l_i, r_i]$ 中的数 x 变成 $(a_i x + b_i) \bmod m$ 。
- 询问: 如果对原序列进行 $[s_j, t_j]$ 中的这些变化, 第 p_j 个数会变成多少。

constriction

$1 \leq n \leq 10^6, 1 \leq q \leq 10^5$.

uoj46 玄学

description

有一个序列 $\{a_i\}$ ，支持 q 次如下两种操作：

- 加入一个变换：将区间 $[l_i, r_i]$ 中的数 x 变成 $(a_i x + b_i) \bmod m$ 。
- 询问：如果对原序列进行 $[s_j, t_j]$ 中的这些变化，第 p_j 个数会变成多少。

constriction

$1 \leq n \leq 10^6, 1 \leq q \leq 10^5$.

solution

合并两个区间的信息需要与 $O(\text{区间长度})$ 的时间复杂度。

不过这里询问的信息是可分割的，所以只需要在 $O(\log n)$ 个线段树节点上维护的信息里做一次二分就行了。

线段树合并

把两棵线段树按对应位置叠加起来。

分析一下线段树合并的复杂度。考虑线段树的每个节点，假设这个节点代表的区间内共包含 x 个元素，那么这个节点在合并时至多只会被访问 $x - 1$ 次，再结合线段树的深度为 $O(\log n)$ ，不难分析出若需要合并的元素共有 m 个，则总时间复杂度为 $O(m \log n)$ 。

zjoi2019 语言

description

给一棵 n 个点的树和树上的 m 条路径，问有多少对 (i, j) 满足这两点同时被至少一条路径覆盖。

constriction

$1 \leq n, m \leq 10^5$.

zjoi2019 语言

description

给一棵 n 个点的树和树上的 m 条路径，问有多少对 (i, j) 满足这两点同时被至少一条路径覆盖。

constriction

$1 \leq n, m \leq 10^5$.

solution

对每个点求所有覆盖该点的链的链并大小，加起来除以 2 就是答案。

树上差分，即对于链 (x, y) 在 x 和 y 处加入这两个点，在 $fa_{lca(x,y)}$ 处删去（注意判出现次数）。线段树合并即可，复杂度 $O(n \log n)$ 。

bzoj2957 楼房重建

description

有 n 栋楼房，第 i 栋楼房可以视作一条连接 $(i, 0)$ 和 (i, H_i) 两点的线段。有 m 次修改操作，每次操作为修改第 x 栋楼房的高度为 y ，在每次修改后求出从 $(0, 0)$ 点能看到多少栋楼房。

constriction

$1 \leq n, m \leq 10^5$.

bzoj2957 楼房重建

description

有 n 栋楼房，第 i 栋楼房可以视作一条连接 $(i, 0)$ 和 (i, H_i) 两点的线段。有 m 次修改操作，每次操作为修改第 x 栋楼房的高度为 y ，在每次修改后求出从 $(0, 0)$ 点能看到多少栋楼房。

constriction

$1 \leq n, m \leq 10^5$.

solution

把 H_i 除以 i ，问题变成求序列从前往后的单调栈长度。

在线段树每个节点上维护一个 mx 和 len 表示这个区间内单调栈的末尾元素和长度。实现一个函数 $query(x, v)$ ，表示前面的单调栈已有最大元素 v ，在后面接上线段树节点 x 后单调栈的增量（增加多少元素、最大元素变成了多少），可以发现若维护出了上述的 mx 和 len ，一次 $query(x, v)$ 只会向 x 的一个子树递归。

考虑维护 mx 和 len 。发现每次 $pushup$ 时都需要向右子树调用一次 $query$ ，因此总时间复杂度是 $O(m \log^2 n)$ 。

线段树势能分析

一般的线段树修改/查询操作都是定位到某个区间 $[l, r]$ 对应的 $O(\log n)$ 个线段树节点后就返回。如果不严格做到这一点，时间复杂度又会变得如何呢？

线段树势能分析一般采用对每个节点定义势能函数的方式来分析时间复杂度。可以粗略地认为，每次的非正常操作（定位到对应节点后不返回）都会导致势能降低，那么总时间复杂度将不会超过总势能累积量。

bzoj5312 冒险

description

一个长度为 n 的序列 $\{a_i\}$ ，支持单点修改，区间与/或一个数，求区间最大值。

constriction

$1 \leq n, m \leq 2 \times 10^5$.

bzoj5312 冒险

description

一个长度为 n 的序列 $\{a_i\}$ ，支持单点修改，区间与/或一个数，求区间最大值。

constriction

$1 \leq n, m \leq 2 \times 10^5$.

solution

定义一个线段树节点的势能为，这个节点表示的区间内所有数在多少个二进制位上不全相同。当一次修改定位到线段树上某个节点时，若这次修改对整个区间的影响相同则直接打加法标记，否则必然引起势能降低，可以暴力递归。该算法的时间复杂度是 $O((n + m \log n) \log \max\{a_i\})$ 。

李超树

支持区间插入一次函数，单点求最值。基本原理是在每个线段树节点上保存使 mid 取到最优的那个一次函数，当某个一次函数被“击败”时，比较两者的斜率，将被击败的一次函数向可能取到更优的一侧递归插入。

有时可以用于代替动态凸包（维护斜率优化等）。

splay

每次从根访问到一个节点后必须将其 splay 方可保证复杂度。

没有深度 $O(\log n)$ 的性质，复杂度均摊 $O(\log n)$ 。由于是均摊复杂度因此无法实现可持久化。

如果想要提取大小为 n 的 splay 上 $[l, r]$ 区间，可以将 $l-1$ 旋到根，将 $r+1$ 旋到根的右儿子，这样 $[l, r]$ 的所有点就根的右儿子的左儿子这棵子树。可以在左右设置哨兵节点从而不加特判地提取区间。

复杂度的证明咕咕咕了。

无旋 treap

对每个节点赋一个随机权值，对序列按随机权值构笛卡尔树，这就是 treap 的形态。

treap 上点 i 是点 j 的祖先的概率是 $\frac{1}{|i-j|+1}$ ，即点 i 的期望深度为

$\sum_{j=1}^n \frac{1}{|i-j|+1} = O(\log n)$ ，因此 treap 的复杂度为期望 $O(\log n)$ 而非均摊，可以实现可持久化。

无旋 treap 只需要支持两种操作：分裂 (split)、合并 (merge)。(见板书)

实际上 treap 可以不记随机权值，在合并两棵子树 x 和 y 时，以 $\frac{sz_x}{sz_x + sz_y}$ 的概率将 x 作为新树根。建议在可持久化 treap 中使用这个技巧（因为可能会复制节点，过多相同权值的节点会导致复杂度出问题）。

codeforces702F T-shirts

description

有 n 种 T-shirts, 第 i 种有价格 c_i 和品质 p_i 。

有 m 个人要来买 T-shirts, 第 j 个人有 v_j 块钱。

每个人的策略都是相同的: 在自己能买得起的所有 T-shirts 中, 买一件品质最高的, 重复此过程直到买不起任何一件 T-shirt。

每种 T-shirt 每人限购一件, 问最终每个人买到了多少件 T-shirts。

constriction

$1 \leq n, m \leq 2 \times 10^5, 1 \leq c_i, p_i, v_j \leq 10^9$.

codeforces702F T-shirts

description

有 n 种 T-shirts, 第 i 种有价格 c_i 和品质 p_i 。

有 m 个人要来买 T-shirts, 第 j 个人有 v_j 块钱。

每个人的策略都是相同的: 在自己能买得起的所有 T-shirts 中, 买一件品质最高的, 重复此过程直到买不起任何一件 T-shirt。

每种 T-shirt 每人限购一件, 问最终每个人买到了多少件 T-shirts。

constriction

$1 \leq n, m \leq 2 \times 10^5, 1 \leq c_i, p_i, v_j \leq 10^9$.

solution

对人按钱数维护平衡树, 按 p_i 从大到小枚举 T-shirts, 问题变成每次给钱数 $\geq c_i$ 的人答案加 1, 钱数减 c_i 。把所有人按钱数分成 3 类: $[0, c_i), [c_i, 2c_i), [2c_i, +\infty)$, 只需要归并前两部分后与第三部分拼接。直接暴力枚举第二部分插入第一部分, 可以发现每个人只会被暴力插入 $O(\log v_j)$ 次, 因此总时间复杂度是 $O(n \log n \log \max\{v_j\})$ 。

替罪羊树

取一个平衡系数 $\alpha \in (\frac{1}{2}, 1)$ ，当平衡树上一个点 x 及其左右儿子 ls, rs 满足 $\max(sz_{ls}, sz_{rs}) > \alpha sz_x$ 时，暴力 $O(sz_x)$ 地重构整棵子树。

可以证明替罪羊树的复杂度是均摊 $O(\log n)$ ，同时值得注意的一点是，替罪羊树的树高不超过 $\log_{\frac{1}{\alpha}} n$ 。一般来说 α 取 $(0.6, 0.7)$ 之间。

重量平衡树

重量平衡树一词是陈立杰在其 2013 年候选队论文中提出的，大家都不知道这个词具体是什么意思。

一种典型的应用是，实现一个支持 $O(\log n)$ 插入， $O(1)$ 实现比较两个元素相对大小（如后缀排序）的数据结构，传统方式是 $O(\log n)$ 地在平衡树上找到两者的位置比较，可以对平衡树的每个节点维护一个实数区间 $[l, r]$ ，根节点的区间为 $[0, 1]$ ，每个节点的权值为 $val_x = \frac{l+r}{2}$ ，且其左右子树的区间分别为 $[l, val_x]$ 与 $[val_x, r]$ ，可以发现这样比较两元素的大小就变成了比较两元素对应权值的大小，唯一的问题在于对精度的要求是关于深度的指数级别，使用 treap 或替罪羊树这种保证深度的平衡树。

实链剖分

实链剖分中，它依据的是每个点，它的实儿子为子树中最后被访问的结点所在子树的根。

用 splay 维护每条实链，实链剖分的关键操作是 `access`，把一个点到根的路径变成一条实链。

实链剖分还可以拓展到动态的树上，支持换根、link、cut 等很多操作。

另外 Link-Cut Tree 也可以维护子树信息和子树标记，只要把实儿子和虚儿子的信息或标记分开维护即可。

sdoi2017 树点涂色

description

一棵 n 个点的有根树，初始时每个点有一个不同的颜色，定义一条路径的权值为这条路径上不同的颜色种数。要求支持三种操作：

- 把 x 到根路径上所有点染上同一种从未出现过的颜色。
- 求 x 到 y 路径的权值。
- 求 x 子树内所有点到根路径的权值最大值。

constriction

$$1 \leq n, m \leq 10^5.$$

sdoi2017 树点涂色

description

一棵 n 个点的有根树，初始时每个点有一个不同的颜色，定义一条路径的权值为这条路径上不同的颜色种数。要求支持三种操作：

- 把 x 到根路径上所有点染上同一种从未出现过的颜色。
- 求 x 到 y 路径的权值。
- 求 x 子树内所有点到根路径的权值最大值。

constriction

$$1 \leq n, m \leq 10^5.$$

solution

很容易发现第一种操作就是 LCT 的 `access`。对 LCT 的每一条实链，在链顶位置打上 $+1$ 标记，这样一个点到根路径的权值就是这个点到根路径上的标记数。

第二种操作的答案是 $val_x + val_y - 2val_{lca(x,y)} + 1$ 。

用线段树维护每个的权值，即可支持区间查询最大值。

bzoj3514 Codechef MARCH14 GERALD07 加强版

description

n 个点 m 条边的无向图, q 次询问保留图中编号在 $[l, r]$ 的边的时候图中的连通块个数。
强制在线。

constriction

$1 \leq n, m \leq 10^6$.

bzoj3514 Codechef MARCH14 GERALD07 加强版

description

n 个点 m 条边的无向图， q 次询问保留图中编号在 $[l, r]$ 的边的时候图中的连通块个数。强制在线。

constriction

$1 \leq n, m \leq 10^6$.

solution

连通块个数等于点数减去树边的数量。

按编号从小到大加入边并维护最大生成树，对第 i 条边求出它加入后会挤掉第 pre_i 条边，特别的， $pre_i = 0$ 表示它没有挤掉任何一条边。此时第 i 条边会使所有 $pre_i < l \leq i \leq r$ 的询问的答案减 1。二维数点即可。

codeforces1109F Sasha and Algorithm of Silence's Sounds

description

给出一个 $n \times m$ 的网格图，每个格子上有一个数，形成一个 $n \times m$ 阶排列。
求有多少个区间 $[l, r]$ ，使得这个区间内的所有数所在的格子网格图上构成一棵树。

constriction

$1 \leq n \times m \leq 10^6$.

codeforces1109F Sasha and Algorithm of Silence's Sounds

description

给出一个 $n \times m$ 的网格图，每个格子上有一个数，形成一个 $n \times m$ 阶排列。
求有多少个区间 $[l, r]$ ，使得这个区间内的所有数所在的格子网格图上构成一棵树。

constriction

$1 \leq n \times m \leq 10^6$.

solution

树的限制可以拆成无环 + 连通块个数为 1。

无成环的限制可以用 LCT + 单调指针解决。

要求连通块个数为 1，由于构成一棵树，那么连通块个数就等于点数减边数。由于点数已知，所以只需要维护区间边数就行了。

稍加转化可以变成线段树区间加区间求 1 的个数（等价于求最小值及其个数）的操作。

虚树

把需要用到的点和任意两点（实际上只需要 dfs 序相邻两点）的 LCA 拿出来建树。

由于把原树中的一条路径缩成了一条边，所以需要想办法快速维护路径信息。

sdoi2011 消耗战

description

给一棵树，边带权。

有一些询问，每次询问给出 k 个点，要求删除边权和最小的边集使得从 1 号点出发不能到达这 k 个点中的任意一个。

constriction

$1 \leq n, \sum k \leq 10^6$.

sdoi2011 消耗战

description

给一棵树，边带权。

有一些询问，每次询问给出 k 个点，要求删除边权和最小的边集使得从 1 号点出发不能到达这 k 个点中的任意一个。

constriction

$1 \leq n, \sum k \leq 10^6$.

solution

对给出的 k 个点建虚树，简单 dp 即可。需要求出树上路径最小值，不过实际上这个路径最小值可以用到根路径最小值代替。

hnoi2018 毒瘤

description

给一张 n 个点 m 条边的图，求独立集个数 $\bmod 10^9 + 7$ 。

constriction

$1 \leq n \leq 10^6, n - 1 \leq m \leq n + 15$.

hnoi2018 毒瘤

description

给一张 n 个点 m 条边的图，求独立集个数 $\text{mod } 10^9 + 7$ 。

constriction

$1 \leq n \leq 10^6, n - 1 \leq m \leq n + 15$.

solution

暴力做法是 $O(2^{m-n+1}n)$ 枚举非树边的选取后做一遍树 dp。

可以发现只需要把这些非树边涉及到的点拿出来建虚树后做 dp 就行了，由于每次虚树的形态是一样的，所以可以预处理出虚树上每条边从 $dp_{v,0/1}$ 转移到 $dp_{u,0/1}$ 的转移系数，复杂度 $O(n + 2^{m-n+1}(m - n + 1))$ 。

一个分块套路

n 个元素被分成了 m 个集合，每次询问两个集合之间的信息。

假设询问的集合是 x 和 y ，如果信息支持 $O(\min(sz_x, sz_y))$ 地查询，那么直接枚举较小的集合中的每个元素查询，这样的时间复杂度就是 $O(n\sqrt{n})$ 的。

经典问题

description

给一个序列 $\{a_i\}$ ，支持区间加下标，区间加常数，查询区间最大值。

constriction

$$1 \leq n, m \leq 10^5.$$

经典问题

description

给一个序列 $\{a_i\}$ ，支持区间加下标，区间加常数，查询区间最大值。

constriction

$1 \leq n, m \leq 10^5$.

solution

分块维护凸包。

Wannafly 挑战赛 12F 小 H 和圣诞树

description

一棵 n 个点的树，每个点有个颜色 col_i ，每条边有个边权。
 q 次询问，每次给两种颜色 a, b ，求 $\sum_{col_x=a, col_y=b} dist(x, y)$ 。

constriction

$1 \leq n, m \leq 10^5$.

Wannafly 挑战赛 12F 小 H 和圣诞树

description

一棵 n 个点的树，每个点有个颜色 col_i ，每条边有个边权。
 q 次询问，每次给两种颜色 a, b ，求 $\sum_{col_x=a, col_y=b} dist(x, y)$ 。

constriction

$1 \leq n, m \leq 10^5$.

solution

设阈值 S ，对于一种出现次数超过 S 的颜色，对整棵树做两遍 dfs ，对每个点求处到这种颜色所有点的距离和，查询时若一种颜色的出现次数超过 S ，则直接暴力枚举较少的一种颜色统计答案。否则两种颜色都出现不超过 S 次，把这不超过 $2S$ 个点拿出来建虚树后 dp （最好对多组询问一起建虚树做 dp ），时间复杂度为 $O(\frac{n^2}{S} + n\sqrt{n} + nS\log n)$ ，当 S 取 $O(\sqrt{\frac{n}{\log n}})$ 时复杂度为 $O(n\sqrt{n\log n})$ 。（默认 n, m 同阶）

谢谢大家！