

区间数据结构入门

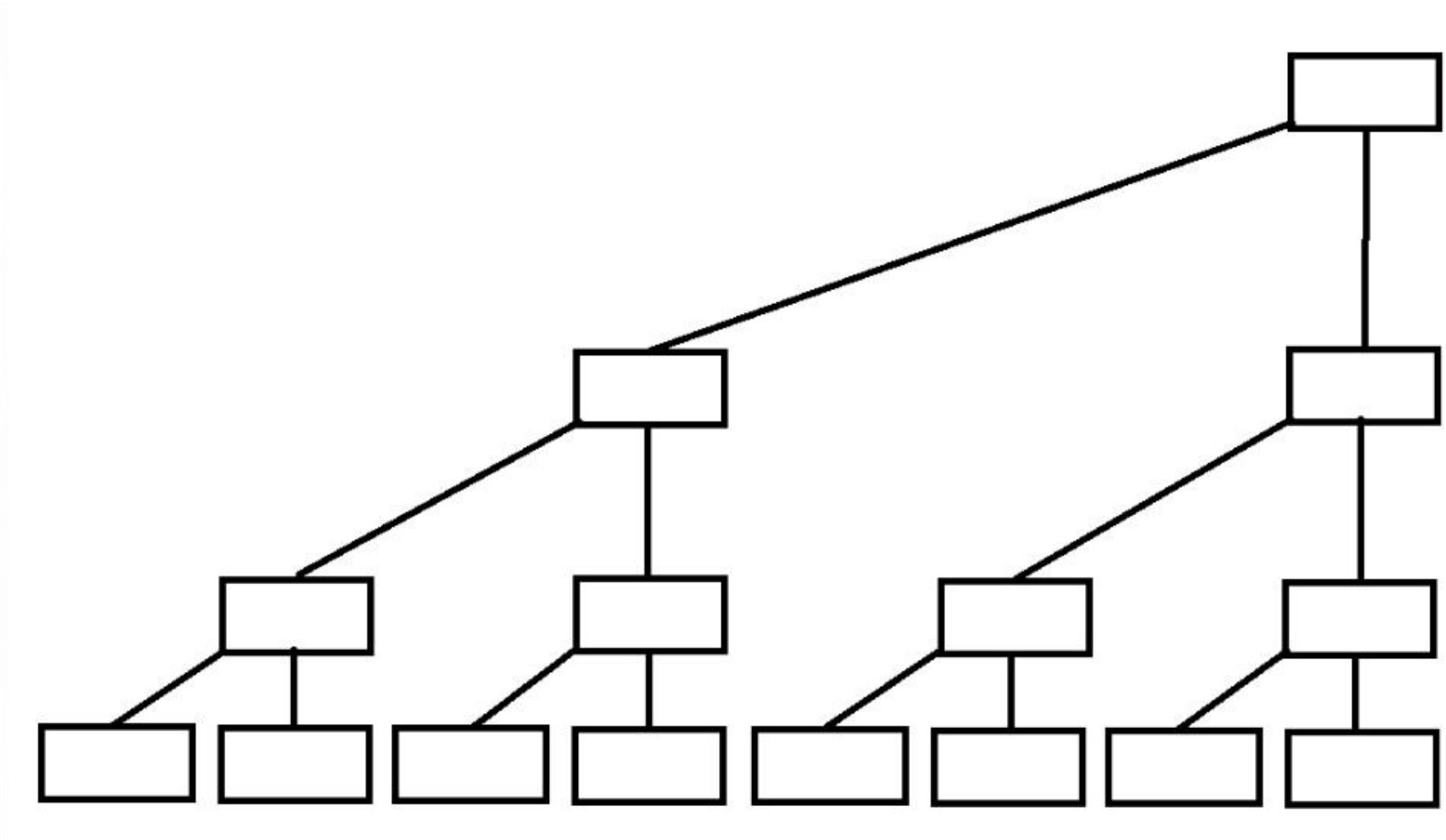
- 一个长为 n 的数组, 多次询问区间和。
- $n \leq 10^5$

- 记 S_i 为前缀和数组, 即 $S_i = S_{i-1} + a_i$, $S_r - S_{l-1}$ 即为所求。

- 1.单点修改
- 2.区间求和

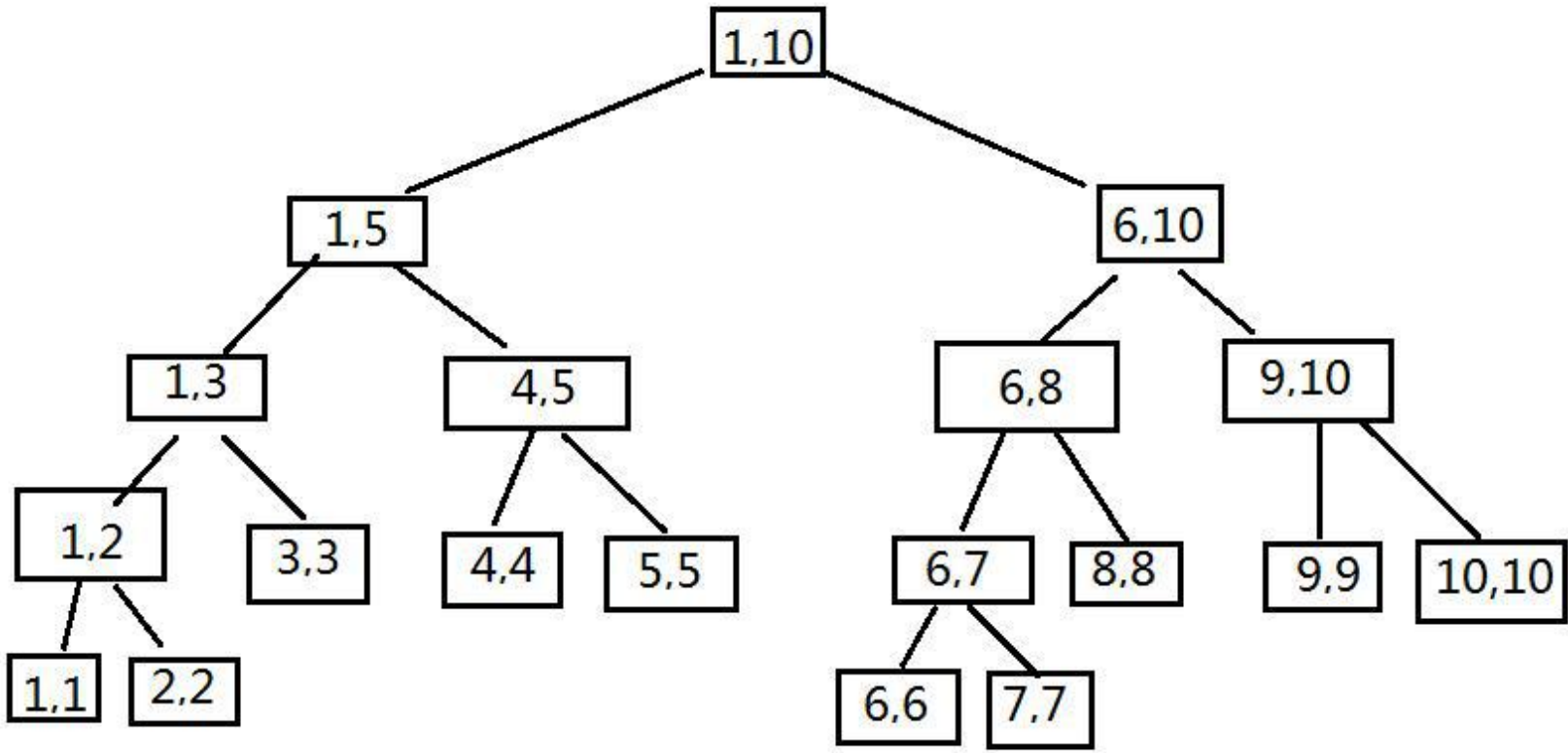
- 无法快速维护前缀和数组了。
- 分块？若干个元素一起维护，询问时变为若干个块的和，以及两边零散部分的和。
- $O(1)$ 修改, $O(\sqrt{n})$ 查询。

树状数组



- 2号点存放 $[1,2]$ 的和, 4号点存放 $[1,4]$ 的和。
 - 记 $lowbit(x)$ 表示 x 的二进制最后一个1, 则 x 号点存放的是从 $x - lowbit(x) + 1$ 到 x 的和。
 - 因此询问时每次 $x -= lowbit(x)$ 。
 - 修改时每次 $x += lowbit(x)$ 。
-
- loj130

线段树



- 每个节点存它对应的区间和即可。
- 修改时只需修改 $O(\log n)$ 个区间。
- 询问时只会定位出 $O(\log n)$ 个区间。

- 1.区间加
- 2.区间求和

- 懒标记：
- 修改时只在定位出的区间上修改，并作标记，表示子树内的部分也会被它影响，但暂不修改。
- 在必要的时候，下传懒标记，修改其子节点。

- luogu3372

- 1.区间加
- 2.区间乘
- 3.区间求和

- 两种懒标记。
- (x,y) : 先乘 x 后加 y 。
- luogu3373

- 1.单点修改
- 2.区间最大子段和

- 每个节点维护三个值 $Lmax, Rmax, Smax$ 表示前缀最大子段和, 后缀最大子段和, 区间最大子段和。
 - 前缀最大子段和: 左儿子的前缀最大子段和或左儿子的和+右儿子的前缀最大子段和。
 - 区间最大子段和: 左右儿子的答案或左儿子的后缀最大子段和+右儿子的前缀最大子段和。
-
- luogu SP1716

- 1.单点修改
- 2.求区间 $[l_i, r_i]$ 内最靠近左端点的大于 x_i 的数。

- 维护区间最大值。
- 询问时二分位置 m , 求 $[l_i, m]$ 的最大值与 x_i 的大小关系。
- $O(n \log^2 n)$

线段树上二分

- 若需要往两边递归:
- 若左边有答案, 则不往右边求, 否则递归右边
- 换句话说, 先取出定位出的 $O(\log n)$ 个区间, 找到第一个最大值大于 x_i 的区间, 然后再到这个区间内查找答案, 由于是线段树的完整区间, 答案易求。
- $O(n \log n)$

动态开点

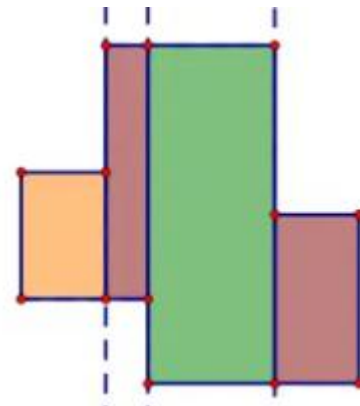
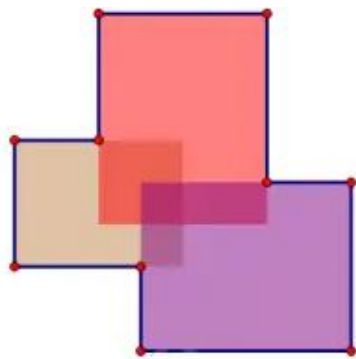
- 不建立整棵树，只建立需要的节点。
- 每次多 $O(\log n)$ 个节点。

权值线段树

- 每个节点表示权值区间 $[l,r]$ 的信息。

- 求 n 个矩形的面积并。

- 扫描线。



- 区间加线段，删线段，求覆盖长度。

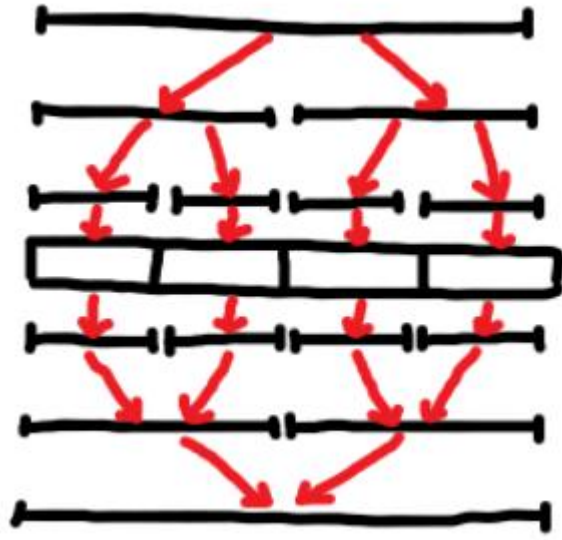
- luogu5490

- 求 n 个矩形的并的周长。

- 1.将 x, y 分开考虑。
- 累加每次覆盖长度的差。
- 2.同时考虑
- 额外统计区间内的线段数量，它们会向上延伸。

- n 个点, 三种边:
- 1. u 到 v , 长为 w
- 2. u 到 $[l,r]$ 中的每个点, 长为 w
- 3. $[l,r]$ 中的每个点到 v , 长为 w
- 求单源最短路。

- 线段树优化建图。

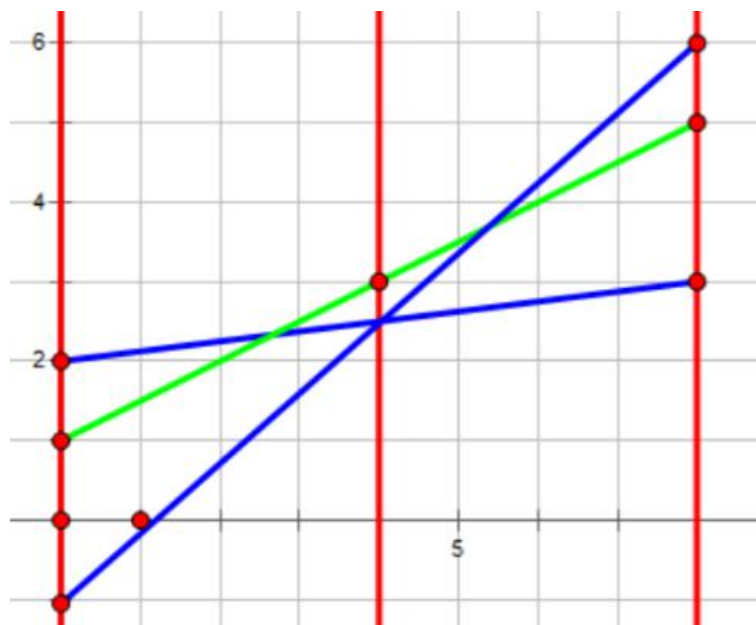


- 1.在平面上加入一条不与 y 轴平行的线段
- 2.求线段与 $x = k$ 的直线交点的最大值

李超树

- 每个线段树节点保存中点处最大的线段。
- 对于其它线段，最多只能往一边递归。
- 分成 $O(\log n)$ 个区间，每个区间又都会下推到叶子。
- $O(n \log^2 n)$

- luogu4097



线段树合并

- 线段树中相同的位置，若两树有一个为空，则返回另一个，否则递归合并左右两边。
- 复杂度？
- 递归时都是两树都有的节点，相当于重复的节点合并了，减少一个。
- 节点总数是 $O(n \log n)$ 的。

- 一棵树，多次操作，会在一条链上的每个节点都存入一个数。
- 所有操作完成后，求每个节点中数量最多的数。

- 设1为根，树上差分，变为1到u，1到v，减去1到lca，减去1到fa[lca]，每个节点的数变为其子树内的数。
- luogu4556

可持久化线段树

- 记录线段树修改的每个版本。
- 修改只会改变 $O(\log n)$ 个节点，新建这些节点即可。

- 1.询问 $[l,r]$ 内的第 k 小数。

- 从 $\text{root}[i]$ 推到 $\text{root}[i+1]$ 。

- luogu3834