二维序列问题

刘承奥 (CommonAnts)

问题引入

- 问题.（实数堆）
- 有一个实数集合S，需要支持：插入一个数、删除一个数、查询集合最大值。

- 如果有解决这个问题的数据结构，则可以用 $O(n)$ 次插入、删除、查询给n个实数排序。
- 因为基于比较的排序的比较次数至少是$\Theta(n \log n)$，所以插入、删除、查询三个操作至少有一个平均每次不低于$\Theta(\log n)$。
  - 比较次数至少是$\Theta(n \log n)$：可能的顺序有 $n!$ 种，每次比较只有2种结果，所以至少有一种结果保留了1/2可能的顺序。要保证找到正确顺序，比较次数至少为 $\log_2 n! = \Theta(n \log n)$。类似地，可以证明随机输入对应的期望时间复杂度至少是$\Theta(n \log n)$ 。

问题引入

- 问题.（实数最值）
- 有一个实数集合S，需要支持：插入一个数、查询集合最大值。
- 只需要维护全局最大值即可。单步严格 $O(1)$

- 问题.（实数栈最值）
- 有一个实数栈S，需要支持：插入一个数、删除栈顶的数、查询全局最大值。
- 只需要维护每个元素到栈底之间的最大值（也就是它刚插入的时候的全局最大值）即可。单步严格 $O(1)$
- 也可以理解为带撤销的实数最值。

问题引入

- 问题.（实数队列最值）
- 有一个实数队列S，需要支持：
- 插入一个数到队头、删除队尾的数、查询全局最大值。

- 因为实数堆是做不到单步平均 $O(1)$ 的，考虑这个问题相比实数堆有什么特殊性质。
- 区别在于，只会删除队尾的数，也就是剩余的插入时间最早的数。
- 可以注意到，如果数x_1的插入时间为t_1，数x_2的插入时间为t_2，且$t_1 < t_2$，那么x_1一定比x_2删除早。如果$x_2 > x_1$，则此后x_1不可能成为全局最大值。
- 所以，对于某个数x，只要存在一个数值比它大的数插入时间也比它大，那么这个数x就不可能再是最大值。

问题引入

- 问题.（实数队列最值）
- 有一个实数队列S，需要支持：
- 插入一个数到队头、删除队尾的数、查询全局最大值。

- 对于某个数x，只要存在一个数值比它大的数插入时间也比它大，那么这个数x就不可能再是最大值。
- 我们只需要存储那些可能是最大值的数。也就是说，不存在一个数值比它大的数插入时间也比它大。用平面上的点(t, x)表示时刻t插入的值为x的数，那么，一个数有可能成为最大值**当且仅当它的右上方没有其它任何点**。
- 我们只需要存储右上方没有其它任何点的点。

问题引入

- 问题.（实数队列最值）
- 有一个实数队列S，需要支持：
- 插入一个数到队头、删除队尾的数、查询全局最大值。

- 考虑原操作对“所有右上方没有其它点的点”集合M的影响：
- M满足：M中元素按t递增排序时，x一定递减。
- 插入一个数：插入一个横坐标t最大的点(t,x)，**原本在M中的所有纵坐标不大于x的点应该被删除**。
- 删除一个数：删除一个横坐标t最小的点，对M中其余的点无影响。
- 查询全局最大值：查询M中纵坐标x最大的点，可以发现，这也是M中横坐标t最小的点。

单调队列？

- 问题.（实数队列最值）
- 有一个实数队列S，需要支持：
- 插入一个数到队头、删除队尾的数、查询全局最大值。

- 所以我们用一个双端队列Q按插入时间t从小到大存储M的信息即可。
- 每个输入的元素最多插入到Q一次，从Q删除一次，所以总的操作次数为**均摊**$O(1)$。

- 因为Q中的元素按t单调递增，按x单调递减，数据结构Q被称为单调队列。

单调队列

- 单调队列只能查询全局最大值吗，它维护了什么信息？

- 插入一个横坐标最大的点，删除一个横坐标最小的点，
- 维护所有**右上方没有其它点的点**。这些点的横坐标单调递增，纵坐标单调递减。

- 对称地，改为最小值可以维护右下方没有其它点的点。

单调栈？

- 问题.
- 有一个实数栈S，需要支持：
- 插入一个数到栈顶、删除栈顶的数、查询有多少个数上方没有比自己大的数。

- 如果这个算法是$O(n)$的，则仍然可以做期望$O(n)$时间排序！
- 做法：首先随机洗混序列，对前$0.9n$个元素递归排序，然后把这$0.9n$个元素按顺序插入栈。对于后$0.1n$个元素，依次插入并立刻删除，就可以得到前$0.9n$个元素中有多少比它小。然后对每个间隙里的元素再排序。可以证明，任何输入分布对应的期望时间复杂度为$O(n)$。这和比较排序时间复杂度的界矛盾！
- 类似地，这个问题，以及更一般的，给定一个集合，多次询问小于等于某个数的个数问题，渐进均不优于排序的时间复杂度。

单调栈

- 问题.
- 有一个数列$\{a_i\}$和一个区间$[L,R]$，需要支持：
- $R++,R--$，查询只考虑区间内元素的情况下，有多少个数右边没有比自己大的数。

- 维护所有**右上方没有其它点的点**。这些点的横坐标单调递增，纵坐标单调递减。
- 插入(i,a_i)和单调队列一样，移除栈里（横坐标必定小于i）所有纵坐标小于a_i的数。
- 删除？如果直接撤销操作，则一个数可能被多次插入删除。均摊时间复杂度错误。
- 对于每个点，我们可以存下它插入的时候在单调栈上相邻的点。也就是，它左侧第一个纵坐标比它大的点。这个点是固定的。

单调栈

双端单调队列

二维偏序

上升子序列

下标-值

链和反链

Dilworth定理及其对偶

- 对于有限个元素的偏序：
- **最大反链等于最小链覆盖。**
- **最大链等于最小反链覆盖。**

- 此处，最小链覆盖指的是选择最少个数的链使得每个元素至少在一个链内。

网格图和对偶图

状态空间

笛卡尔树对偶

区间二维偏序

最小区间覆盖

区间调整法例题