

《cashier》设计报告

贝小辉

【题目背景】

本题是第一试的第一题，也是比较简单的一道题，重点考察大家对于基本数据结构的掌握与应用。

【标准算法】

题目描述并不难理解，简单来说就是请你设计一种数据结构满足如下 4 种操作：

1. 向集合中插入一个数；
2. 将集合中所有的数都加上一个值；
3. 将集合中所有的数都减去一个值，并将所有低于 \min 的数从集合中删除掉（ \min 是事先给定的一个固定的数）；
4. 查找集合中第 k 大的数。

为了便于估计算法的时间效率，我们对数据规模进行一下说明：插入元素的命令数目为 N ，修改集合内元素的值（即加上一个值或减去一个值）的命令数目一共为 M ，查找的命令数目为 F ，由题目中的约定可以得到 $N \leq 100000$, $M \leq 100$, $F \leq 100000$ 。

总体看来，由于操作中包含了插入、删除元素，以及查找第 k 大的数，能够实现这些操作的最佳选择就是静态排序二叉树，每种操作的时间复杂度都为 $O(\log(\text{Limit}))$ ，其中 Limit 为集合中元素的范围，在本题中为 100000。

但是还有一种操作就是将集合内的所有元素都加上或减去一个值，如果单纯的按照题目要求对数据进行改动，则每一步这样的操作所需的时间复杂度为 $O(N \log N)$ （因为需要重建二叉树）， M 次操作的时间复杂度就为 $O(MN \log N)$ ，这是我们无法承受的。注意到这种操作并不会改变集合内元素的大小关系，所改变的只是已经在集合内的元素与以后将要加进来的元素间的关系，这样我们就可以只改变以后要加进来的元素而不改变当前已有的元素。具体做法为设立一个表示改变量的变量 δ ，如果遇到将集合内所有元素加 a 的操作，不改变

集合内元素的值而令 $\text{delta} = \text{delta} + a$ ，同理遇到将集合内所有元素减 a 的操作则令 $\text{delta} = \text{delta} - a$ ，同时将集合内所有小于 $\text{min} - \text{delta}$ 的元素删除掉。在以后再遇到插入一个值为 x 的元素的操作时，只要将 x 修改为 $x - \text{min}$ 后再插入集合即可。这样就把修改集合内元素值的操作的时间复杂度降到了 $O(1)$ ，插入一个元素和查找集合中第 k 大的数的时间复杂度仍然为 $O(\text{Log}(\text{Limit}))$ ，由于每个元素最多只能被删除一次，所以删除元素的操作所需总的时间复杂度为 $O(N\text{Log}(\text{Limit}))$ ，故算法总的时间复杂度为 $O(N\text{Log}(\text{Limit}) + N\text{Log}(\text{Limit}) + M + F\text{log}(\text{Limit}))$ ，但要注意到这时集合中元素的范围 Limit 已不再是 100000 了，修改集合内元素值的超作会影响到后面插入的元素的范围。由于每次集合内元素改变的值不超过 $\text{PerLimit}=1000$ ， delta 最大即为 $M * \text{PerLimit}$ ，故 $\text{Limit} = 100000 + M * \text{PerLimit} \leq 200000$ 。

【其他算法】

- 用数组表示集合，每次插入操作将元素插到数组末尾。修改元素时修改集合内的所有元素，并在将集合内所有元素减去一个值时将元素由大到小排序，再将所有小于 min 的元素删除。查找第 k 大的元素时用平均复杂度为 $O(N)$ 的算法。
 - ✓ 时间复杂度： $O(N + M * N * \text{Log}(N) + F * N)$
 - ✓ 预计得分：30 分
- 用数组表示集合，在执行每一步操作时始终保证数组内的元素有序。
 - ✓ 时间复杂度： $O(N^2 + M * N + F)$
 - ✓ 预计得分：50 分
- 动态排序二叉树。算法与标准算法相同，但数据结构采用动态排序二叉树。
 - ✓ 时间复杂度：当数据为随机数据时与标准算法相同，最坏为 $O(N^2 + N^2 + M + F * N)$ 。
 - ✓ 预计得分：50 分

【测试数据】

测试点编号	N	M	F
1	10	5	5
2	100	50	200
3	10000	5	500
4	20000	100	10000
5	30000	100	20000
6	50000	100	20000
7	50000	100	10000
8	80000	100	50000
9	100000	100	80000
10	80000	100	80000