DZY Loves Data Structures 解题报告

杭州学军中学 王逸松

1 试题来源

学军中学内部集训

2 试题大意

你有n个序列，一开始每个序列里只有一个元素。
每个元素有一个A属性和一个B属性。
有m个操作：

`1 x y val`	修改第x个序列中第y个元素的A属性为val
`2 x y val`	修改第x个序列中第y个元素的B属性为val
`3 x y`	将第y个序列接在第x个序列后，之后的操作中不会出现第y个序列
`4 x l r val`	询问第x个序列的第l到第r个元素中A属性大于val的元素个数
`5 x l r k`	询问第x个序列的第l到第r个元素中A属性第k大的值
`6 x l r Al Ar`	询问第x个序列的第l到第r个元素中，A属性在[Al,Ar]内的元素的B属性的最大值（如果不存在这样的元素，输出0）

强制在线。
保证任意时刻任意元素的A,B属性都在$[1,10^6]$内。

3 数据范围

对于20%的数据，$n \le 2000$, $m \le 5000$。

对于40%的数据，$n \le 70000$, $m \le 200000$。

对于另外20%的数据，$n \le 25000$, $m \le 400000$。

对于100%的数据，$1 \le n \le 250000$, $0 \le m \le 450000$。

时间限制12s

空间限制512MB

4 算法介绍

4.1 算法一

暴力，注意求第k大可以用STL的`nth_element`做到$O(n)$。

时间复杂度$O(nm)$，空间复杂度$O(n)$，可以通过20%的数据。

4.2 算法二

块状链表，每个块维护一棵以A属性为下标的线段树，然后将块的大小设为$O(\sqrt{n}\log n)$，就能做到$O(n\log n+m\sqrt{n}\log n)$的复杂度，可以通过40%的数据。

4.3 算法三

对于“将第y个序列接在第x个序列后”这个操作，可以启发式合并，即将长度较小的序列中的元素暴力插入到长度较大的序列中。

于是问题转化成，给一个序列，要求支持插入一个元素、修改一个元素、、询问区间第k大、询问区间内A属性在一个范围内的B属性最大值。

可以证明每个元素插入的次数是$O(\log n)$的，所以转化后的问题有$O(n\log n+m)$个操作。

对于转化后的问题，可以用重量平衡树（如treap、替罪羊树）套线段树来解决，其中重量平衡树的下标是序列的下标，线段树的下标是A属性的大小。

总时间复杂度为$O(n\log^3 n + m\log^2 n)$，空间复杂度为$O(n\log^2 n)$，可以通过另外20%的数据。

4.4 算法四

考虑“带插入区间K小问题”的另一种做法：对所有权值建一棵线段树，对于每个线段树结点，建一棵下标为序列中的位置平衡树，保存所有权值在这个线段树结点所表示区间内的元素。

询问的时候在权值线段树上走，并在相应的平衡树上询问。

修改的时候更新相应的平衡树。

由于有插入操作，要使用重量平衡树来维护不同元素的先后关系。

这个做法的复杂度是$O(n\log^2 n)$的。

对于原问题的“将第y个序列接在第x个序列后”这个操作，我们需要将这两个序列的权值线段树合并，不妨考虑以下算法：

```
merge(A, B):     // 将线段树A和线段树B合并
    如果A和B的其中一个的平衡树是空，则返回另一个
    将A和B中的平衡树连接
    返回左右儿子分别是merge(A.l, B.l)和merge(A.r, B.r)的线段树
        // 其中A.l表示A的左儿子，A.r表示A的右儿子
```

在这个问题中，还要快速知道不同元素的先后关系。可以对每个元素记录一个整数“标号”，在同一个序列中标号小的元素在标号大的元素前面。一开始每个元素在单独的序列里，标号都是1。每次合并两个序列时，修改长度较小的序列的每个元素的标号，使整个序列的标号连续。容易证明维护所有元素的标号的时间复杂度为$O(n \log n)$。

考虑这个算法的复杂度。

对于一棵线段树A，我们记$size(A)$为A这棵线段树中平衡树非空的**线段树结点**个数。

在合并两棵线段树A, B时，记合并的结果为线段树C，那么这次合并的时间复杂度为$O((size(A) + size(B) - size(C)) \log n)$，同时所有线段树的$size$之和减少了$(size(A) + size(B) - size(C))$。

对于一个修改操作，时间复杂度为$O(\log^2 n)$，同时所有线段树的$size$之和增加了$O(\log n)$。

一开始所有线段树的$size$之和为$O(n \log n)$，可以证明，合并操作和修改操作的总时间复杂度为$O((n + m) \log^2 n)$。

询问操作的复杂度也是$O(\log^2 n)$。

所以总时间复杂度为$O((n + m) \log^2 n)$，空间复杂度为$O(n \log n)$，可以通过所有测试数据。

5 参考程序

见`dlds.cpp`（算法四）