

EX2

王嵘晟 PB1711614

EX2-1

思路：此题是 EX1-1 的升级版，难点在于若干插入语句后有若干查询语句，接下来还可以有若干插入语句。因此不能采用排序后查找的方法，否则会超时。这里我用了二叉树的结构来存储数据，采用平衡二叉树 AVL。建初始空树操作的时间复杂度为 $O(1)$ ，当插入一个结点时，自上而下寻找插入结点位置，并对 AVL 自下而上进行左旋或者右旋操作，所以插入结点的时间复杂度为 $O(\lg n)$ 。二叉树的查找只需比较要找的值的大小和当前结点的值的大小，小于当前值则查找左子树，否则查找右子树，时间复杂度为 $O(\lg n)$ 。

由于查询和输入语句最多有 2000000 条，所以在主函数中用一个 while 循环来反复执行 Insert 和 Find。所以总的算法时间复杂度为 $O(n \lg n)$ 。

EX2-2

思路：由于这道题输入了 n 个最长为 5 的由字母构成的字符串，直接处理过于不方便。所以这里我先做了一次映射：将 a 赋值为 0，以此类推 z=25。这样就可以把任意一个字符串转换成 $0 - 26^5$ 中的任何一个数，用一个数组重新存储这些记录了源字符串映射的数字。这里寻找给定可重复元素个数的子队列采用了滑动窗口这样一个思路。一开始滑动窗口的大小为给定的可重复的元素个数 k ，这时如果滑动窗口内重复元素个数恰好为 k ，则计数值 +1，窗口右端向右移动一个单位。当窗口内重复元素个数大于 k 时，窗口左端右移一个单位，然后记录窗口右端位置，将窗口右端左移 x 个单位使得窗口中不重复元素个数仍为 k 个，这时计数值 + x 。当窗口右端移到数组末尾时，只需移动窗口左端，方法同窗口内元素大于 k 的个情形。

在最坏情况下，如 a b b b b b c c c c c c c c c, $k=2$ 这种情况，按照我的算法时间复杂度将会是 $O(n^2)$ ，不过由于测试数据比较和谐，所以能够通过。

EX2-3

思路：此题依旧采用 AVL 来存储数据，只不过每个节点存储的不再是一个值，而是一个区间。同时进行数据结构扩张，每个结点多开设一个值 MAX 用来记录这个结点及其子树中区间端点的最大值。这时查找算法同样是在平衡二叉树的查找的基础上修改得来的，如果当前结点左子树的最大值大于 x ，则在左子树上查找 x ，反之在右子树上查找 x 。

建空树的过程时间复杂度为 $O(1)$ ，插入结点时间复杂度为 $O(\lg n)$ ，查找的时间复杂度为 $O(\lg n)$ ，由于一共需要执行 1000000 次操作，所以总的时间复杂度为 $O(n \lg n)$ 。

EX2-EX

思路：此题用并查集的思想，先用结构体来存输入的危险品种类与危险系数，然后根据输入的危险系数 α_{ij} 的大小使用快速排序从大到小排序。针对排序号的结构体，利用并查集，先将每个危险品结点的父亲设为自己，然后分为 4 种情况：如果两种危险品没有对立集合，则将彼此设为对立集合，如果两种危险品中其中一个有对立集合，则合并有对立集合和没有对立集合的危险品。并将没有对立集合的危险品的对立集合设为另一个危险品。如果两种危险品都有对立集合，则合并 x 与 y 的对立集合，然后合并 y 与 x 的对立集合。最后如果 x 与 y 在相同集合，那么它们的危险系数一定为所求的最大值。如果上述四种情况都没有，则不存在危险，输出 0。

建立集合时间复杂度 $O(1)$ ， n 次合并 m 次查找最坏情况的时间复杂度为 $O(m\alpha(n))$ ，根据 α 函数性质，此算法最坏时间复杂度为 $O(m)$ 线性时间。