

2017福建省夏令营Day1练习-solution

by zld3794955

2017 年 7 月 17 日

题目名称	工作团队	单词表	数列编辑器
目录	workteam	wordlist	editor
可执行文件名	workteam	wordlist	editor
输入文件名	workteam.in	wordlist.in	editor.in
输出文件名	workteam.out	wordlist.out	editor.out
每个测试点时限	1秒	1秒	1秒
内存限制	256MB	256MB	256MB
测试点数量	10	10	10
每个测试点分值	10	10	10
是否有部分分	否	否	否
题目类型	传统型	传统型	传统型

注意：预计是Win8下Cena评测。

评测机性能足够好且评测时开启了-O2优化

评测时栈空间设为32M，足以满足正常的函数递归需求。

工作团队

【算法思路】

本题的Easy难度就是裸并查集。

Normal难度就是今天上午讲到过的并查集维护集合大小，至于不同集合数则可以通过统计有效合并次数求出。

Hard难度则需要一定的技巧，在树结构的并查集中，我们要使一个点从树中分离出来是一件很麻烦的事。

为了保持树结构的完整，对于每次分离操作，我们为相应的点 u 新建一个节点，表示独立出来的 u ，而原来的节点表示旧的 u 节点，它起到连接节点与保持树结构的作用，在此之后对节点 u 的操作均在新建立的节点上进行即可。

注意分离的时候要维护好原来的集合的 $size$ 值和集合数。

Extra难度的话，我们还需要对集合大小维护一个支持插入删除查询最大值的数集（修改=删除+插入），这种方法在今天早上用堆维护中位数的时候讲过，当然用C++STL的`multiset`也可以。



单词表

【算法思路】

对于这类在一堆字符串中两两的公共前缀类问题，我们一般从前缀出现的次数入手。

显然，某两个字符串的最长公共前缀（即 $S1$ 串）一定在这些字符串中作为前缀出现了至少两次，则我们要找最长的 $S1$ ，只要找到最长的，在这些字符串组中作为前缀出现了至少两次的字符串即可。

利用上午讲的Trie树维护字符串出现次数size的技巧，再维护一下树上每个节点的深度，即可求出最长 $S1$ 的长度。

至于要寻找字典序最小的最长 $S1$ 串，这个可以对Trie树进行一遍DFS回溯遍历求出（明天会讲），如果你还不会DFS，你可以记录一下Trie树上的每个节点对应原字符串集中哪个字符串的哪个前缀，这样就可以获取节点对应的字符串从而进行比较了。

而求有多少对字符串最长公共前缀为所求 $S1$ 串，只需直接数出我们求出的 $S1$ 串作为前缀出现的次数 t ，则 $t \times (t - 1) / 2$ 即为答案。

但对于任意的 S 串，我们上述过程只能求出 S 是多少个字符串对的最长公共前缀的前缀，因此我们还需要分别扣掉 $S + 'a'$ 到 $S + 'z'$ 的结果。

设最长 $S1$ 串长度为 L ，字符串总长为 $sum \leq 500000$ ，那么我们遍历一遍Trie树上的节点复杂度为 $O(sum)$ ，其中深度恰好为 L 的节点个数不超过 sum/L 个，所以进行字符串比较的复杂度也不超过 $O(sum)$ 。

最长公共后缀与前缀类似，将字符串反序建Trie即可。

另外，这题用字符串Hash来解决也是可以的，其基本思想与用Trie类似，只不过字符串hash是直接通过一系列的hash值来求某个前缀出现的次数的，也比较好实现，大家可以自己想想。

数列编辑器

【算法思路】

对于Easy任务，我们直接暴力即可。

对于Normal任务，注意到并不存在修改操作，并且添加和删除元素均是在序列末尾进行的，那么我们利用一个栈来直接维护序列以及序列的前缀和，则区间求和以及修改的复杂度均是 $O(1)$ 的。

Hard任务在Normal任务的基础上加上了单点修改，因此我们可以用栈式的树状数组来维护前缀和。

而对于Extra任务，注意到我们只在光标附近添加和删除元素，并且光标只能一格一格地移动，那么我们可以以光标为界，利用栈式的树状数组维护前半部分的前缀和和后半部分的后缀和即可（栈底为序列两端，栈顶为光标处），插入和删除元素即为在前半部分的栈中插入或弹出元素，而光标移动则相当于把一个部分的栈顶弹出塞入另一个栈，总时间复杂度 $O(n \log n)$ 。

要写Splay、Treap、块状链表的话我不能保证你过。



感觉比我们出的题良心多了。。