

《数据结构》上机报告

2018 年 12 月 8 日

姓名：赵得泽 学号：1753642 班级：电子2班 得分：

研讨题目	<p>1. 二叉排序树</p> <p>它或者是一棵空树；或者是具有下列性质的二叉树：</p> <p>若左子树不空，则左子树上所有结点的值均小于它根节点的值</p> <p>若右子树不空，则右子树上所有节点的值都大于它根节点的值</p> <p>左右子树分别为二叉排序树</p> <p>请思考：</p> <p>(1) 请给出一棵二叉排序树</p> <p>(2) 如何利用该二叉排序树得到一个递升序列？</p> <p>(3) 如何利用该二叉排序树得到一个递降序列？</p> <p>2. 树形结构在文件管理中的应用</p> <p>(1) 怎样利用树形结构来管理文件目录，并且能够将文件和文件夹加以区分；</p> <p>(2) 如何统计一个节点下文件夹和文件的数目；</p> <p>(3) 从目录树的管理上来看，要实现文件夹或者文件的删除、复制、移动，请描述思路。</p> <p>(4) 地址路径和目录树结构怎么映射，给定一个地址路径，怎么实现定位，反之，给定一个节点，获得相应的文件目录；请描述算法思路。</p> <p>3. 有一千万条短信，有重复，以文本文件（ASCII）形式保存，一行一条，请找出重复最多的前十条。</p>
题目一	<p>首先二叉排序树是在构建二叉树的基础上进行的，然后会根据结点值的大小与根节点值进行比较排序即可。</p> <p>(1) 二叉排序树</p> <div data-bbox="753 1473 1042 1711"><pre>graph TD; 8((8)) --> 3((3)); 8 --> 10((10)); 3 --> 1((1)); 3 --> 6((6)); 6 --> 4((4)); 6 --> 7((7)); 10 --> 14((14)); 14 --> 13((13));</pre></div> <p>(2) 利用中序遍历的方法即可得到一个递升序列</p> <p>(3) 利用逆中序遍历的方法即可得到一个递降序列</p>

<p>题目二</p>	<p>树形结构在文件管理中的应用应该还是比较省时和方便的，利用树形结构存储，加快了目录的检索速度；解决了文件重名问题，只要这些文件不在同一个子目录里；便于实现文件保护、加密和共享；可以很好反映现实世界复杂层次结构的数据结合。</p> <p>(1) 区分文件夹和文件：在构建树的时候，可以根据文件夹和文件的特点进行构建，我们可以在一个根目录下存放多个子文件夹，而这些子文件夹又可以根据不同的性质和作用分成多个类，这样就相当于多个子文件夹是根的孩子结点；若是文件的话，文件应该在相应的文件夹下，通过寻找文件夹的子节点（即叶子结点）来寻找文件。将文件夹和文件存储的时候设置标志 tag，是文件就置 0，文件夹就置 1，这样就可以清楚地分开文件夹和文件。</p> <p>(2) 节点下文件夹和文件数目的统计：统计就意味着遍历，所以，为了很好的遍历，构建用孩子表示法，该节点下，将该结点的一级文件夹存放在一个链表中（此即根节点的孩子结点），然后对每个一级文件夹还是按照这样的操作来存储（即存储二、三、四级文件夹……），直到遇到文件，就让它成为孩子结点。最后，文件的数量只要通过读标志 tag 即可得出，文件夹的数量只需要在每次遍历每个节点的时候判断 tag 标志是否为 1 即可，再通过计数即可得出文件夹的数量。</p> <p>(3) 要删除、复制和移动，第一步就是要根据你要删除文件的指令进行查找，在构建的目录树上查找的时候，首先必须在根目录下找到一级文件夹，然后再一级一级进行查找，直到找到所要删除的文件或者文件夹，然后将其所在的节点及子节点全部删除（当然删除文件夹下面的子节点还需要遍历删除）。复制，就是找到某个文件或文件夹，若是文件，则记录其信息并且在你想要插入的地方新建节点，再填充信息即可；若是文件夹，则要将该结节点及所有子节点都进行（遍历）记录（即记录在另一颗新树上），在想要插入的地方直接通过新建节点并且遍历已经记录要复制信息的树即可。对于移动，只需要先复制再删除即可。</p> <p>(4) 首先读入一个字符串，即地址路径，通过读入最先两个‘\’之间的字符串并记录，在根目录的子节点中进行遍历，找到与值之相同的节点，一直重复，直到地址路径字符串结尾两个‘\’之间的字符串，每次都做相同的操作，那么就会最终就会定位到目标文件；相反，若是给出一个具体节点的信息，然后去找路径的话，就需要每次去查找双亲节点，然后压栈，一直重复该操作直到根节点，最后再弹栈，那么就会形成一条到目标节点的地址路径。</p>
<p>题目三</p>	<p>此题为了提高查找效率，应该用字典树实现，字典树又是单词查找树，是树形结构，类似于哈希树、哈希表。常用于文本词频的统计。它的优点是：利用字符串的公共前缀来节约存储空间，最大限度的减少无谓字符串的比较，查询效率高于哈希表。</p> <p>在这个问题中，首先在根节点下，设置 m(即 1000 万条短信中开头字母的种类数，最多为 256)个 ASCII 码值结点，即第一级子节点，这些节点的值分别是 1000 万条短信开头的 ASCII 码，将短信开头 ASCII 码相同的短信归类到同一节点之下，这样就将 1000 万条短信进行了分类；之后分别对根节点的每个分类子节点进行同样的操作，即每个节点下短信的第二个字符</p>

	<p>ASCII 码相同的字符串进行归类，各个不同种类的 ASCII 又构成了 mi 类，即 mi 个子节点……以此类推，直到将所有的字符串分类结束，字典树就建立好了，其实在建立的过程中我们对每个节点的子节点个数进行计数，直到最末尾的叶子结点，那么它记录的个数就是从根节点出发一直按深度遍历到它形成的字符串的个数，即某条相同短信的个数。</p> <p>之后在查找重复最多的前十条的时候，可以设置 MAX，以及标记数组 visited[]（初始化为 0），通过遍历字典树每个叶子节点，求得最大计数个数，并将对应 visited[] 数组中与该最大个数对应下标的元素置 1，防止下次寻找第二大计数个数的时候造成干扰。同时，从该叶子结点向前回溯，并且压栈，直到根节点，之后弹栈，即可得到短信序列。遍历 10 次之后即可找出前 10 大数字以及对应的短信序列。</p>
心得体会	<p>这次研讨课的题目主要是针对树，题目很有研讨的价值，我们通过对这些问题的讨论，充分认识到树在实际生活中应用的广泛性和有效性。</p> <p>通过对二叉排序树的讨论，我认识到查找的高效性，并对以前学过的树的遍历有了更加充分的理解。（即逆中序输出递降序列，中序输出递升序列）还有就是通过对树形结构在文件管理中的应用，充分认识到树形结构带来的层次性和高效性。并且学会了如何通过探讨去解决一个问题、完成一个项目，如何认真而且有效的思考，切中要害。也通过别人的观点看到了自己知识掌握过程中的不足和漏洞，从而不断更新和完善自己知识体系。最后那个题目，我首先自己思考了一番，也想到了这种方法，只不过目前的知识结构不够清晰，不能够系统地表达出来，在看了老师给的提示之后，对字典树也有了初步的认识，也更有利于以后在学习中的研究和探讨。</p> <p>本次研讨课，题目的确很是具有思考的价值，以上便是我们通过研讨以及自己对问题解决提出的一些观点。</p>