

# 数据结构与算法基础课程实验报告

## 实验 1：线性表及其应用

姓名	李大鑫	院系	计算机学院	学号	1170300825	
任课教师	臧天仪		指导教师	臧天仪		
实验地点	格物 207		实验时间	11/10/2018		
实验课表现	出勤、表现得分 10%		实验报告 得分 40%		实验总分	
	操作结果得分 50%					
实验目的：						
<ol style="list-style-type: none"><li>1. 掌握线性表顺序存储结构的特点及线性表在顺序存储结构中各种基本操作的实现。</li><li>2. 掌握线性表链式存储结构的特点及线性表在链式存储结构中各种基本操作的实现。</li><li>3. 重点巩固和体会线性表在链式存储结构上的各种操作和应用。</li></ol>						
实验内容：						
<ol style="list-style-type: none"><li>1. 实现线性表的链式存储结构——线性链表。从文件输入一批整数，建立有序链表（升序），并完成：<ul style="list-style-type: none"><li>• 查找一个指定元素</li><li>• 插入一个给定元素</li><li>• 删除一个指定元素</li><li>• 统计链表的长度</li><li>• 输出线性链表</li><li>• 实现安逆序链表的重建</li></ul></li><li>2. 采用链式结构实现任意多项式的存储，求两个多项式的和。</li></ol>						
实验要求：（学生对预习要求的回答）（10 分）					得分：	
<p>1、实现线性表的链式存储结构——线性链表？</p> <p>构造一个 struct，使用 data 域存储数字，使用一个指针指向下一个节点。</p> <p>构造一个按照从小到大顺序保存的链表，插入的时候将需要插入的值与当前节点的 data 和后一个节点的 data 进行比较，当值处于两个之间的时候进行插入。删除的时候依然遍历链表，找到则删除，删除的时候调用 free，然后链接左右两个节点，需要特殊判断 NULL 的情况。</p> <p>2、我们经常遇到两多项式相加的情况，在这里，我们就需要用程序来模拟实现把两个多项式相加到一起。首先，我们会有两个多项式，每个多项式是独立的一行，每个多项式由系数、幂数这样的多个整数对来表示？</p> <p>使用问题1中的方法进行链表的创建与操作。根据实验要求，我们这里只需要用到链表的insert操作来实现链表的创建。</p> <p>进行多项式相加的思路与归并排序类似，通过第三个链表来存储结果，将需要进行相加的两个链表，对头节点进行比较，如果阶数相等的时候再进行相加，否则将更小的那个加入到新的链表中。因为我的链表创建方法实现的时候都必须调用一次Create然后才能使用Insert进行插入，所以在插入到最新链表的时候都需要进行判断是否需要Create创建链表。</p>						

实验过程中遇到的问题如何解决的？（10 分）（着重从软件调试、质量保证、结果分析方面进行阐述）	得分：
<p>问题 1：</p> <p>1) 读入之后得到的多项式数组维持 <code>exp</code> 降序：写一个 <code>InsertNode</code> 函数来进行元素的插入，比较当前元素值和当前节点元素和下一个节点元素的值，如果需要插入的值处于两者之间，则将该节点插入到两节点之间。需要特殊处理节点头的情况，同时需要意识到后一个节点可能是 <code>NULL</code> 情况，将 <code>NULL</code> 的节点值设置为 <code>inf</code> 最大值，则符合普遍情况。</p> <p>2) 插入、删除、统计、输出通过遍历链表实现</p> <p>3) 逆序重建链表：遍历链表将所有的值域存储到数组中，新建一个链表反向插入数组中的值。</p> <p>问题 2：</p> <p>1) 多项式加法的逻辑实现：用两个指针分别指向两个链表进行比较的最头的节点，比较两个指针指向节点的 <code>exp</code>，将 <code>exp</code> 更大的节点复制添加到和链表的最后，然后移动相应指针，如果两者的 <code>exp</code> 相同，则将两者进行相加然后添加到和链表的最后。</p>	
本次实验的体会（结论）（10 分）	得分：
<p>1) 链表的实现需要特别注意，指针的基础如插入节点、删除节点的操作顺序需要明确。</p> <p>2) 考虑问题需要更加的全面，当两者进行相加之后可能产生多项式的多余项，需要进行处理；同时如果输入的多项式中有多个 <code>exp</code> 相同的项需要对这些相同的项进行合并。</p> <p>3) 在进行链表操作的时候在用到 <code>next</code> 指针的时候需要特别考虑，下一个节点是不是 <code>NULL</code> 的情况。</p> <p>4) 链表的头和链表的尾在进行链表遍历的时候需要给予特别的考虑。</p>	
思考题：（10 分）	
思考题 1：（5 分）	得分：
<p>思考题 1：多项式的求和问题是否可以采用顺序存储方式来实现，为什么？</p> <p>可以。</p> <p>我们可以使用顺序存储方式存储一个节点，节点中存放 <code>exp, coex</code> 信息，如果是数组类型，我们可以将一个节点声明为一个 <code>struct</code>，<code>struct</code> 中存放 <code>exp</code> 和 <code>coex</code>，然后将 <code>struct</code> 用数组来存储。</p> <p>将节点从文件中顺序读入之后，通过一个 <code>sort</code> 排序算法将节点根据 <code>exp</code> 进行排序，这样就可以维持一个多项式数组是维持 <code>exp</code> 降序来存储的。</p> <p>两个已经维持好降序的多项式数组通过类似于“双路排序”合并原理的、与链表实现规则相同的合并方法将两个数组进行加法合并得出第三个数组，这个数组就是我们求的和多项式。</p> <p>需要注意的是：如果需要读入之后就已经得到一个已经按降序排好的多项式，用数组的方法时间复杂度就会下降、代码复杂性就会上升，因此这个时候顺序存储方式就不是一种好的实现方式了。</p>	
思考题 2：（5 分）	得分：

思考题 2：在线性表的实现过程中采用哪种存储结构更好，为什么？

线性表的实现所采用的存储结构需要视具体应用来决定。

顺序存储：

顺序结构存储时，相邻数据元素的存放地址也相邻，要求内存中存储单元的地址必须连续的。

链式存储：

链式存储时，相邻数据元素可随意存放，存储空间必须要保存一部分来存储指针，指针可以指向前面、后面的元素。

两者的实现差别决定了两者各自的优势，顺序存储适合用于索引查值，c 语言中直接下标索引即可，链表实现则需要顺序检查，链表适合实现插入、删除操作，顺序实现则需要移动后续元素。两者的根据值来查找都需要顺序检查，差别不大。

因此当具体选择实现方式的时候，可以根据各自的优点来选择实现方法。

指导教师特殊评语：

指导教师签字：

日期：