数据结构与算法

实验报告

实验题目：线性表基础实验（实现双链表）

姓 名：刘帅

学 号：2020212267

日 期：2021年4月30日

自我评分：【 A 】

自我评分说明：A+，A，B+，B，B-，C，D，分别对应分数95、90、85、80、75、70、60

诚信声明

本人郑重承诺：本实验程序和实验报告均是本人独立学习和工作所获得的成果。尽我所知，实验报告中除特别标注的地方外，不包含其他同学已经发表或撰写过的成果；实验程序中对代码工作的任何帮助者所作的贡献均做了明确的说明，并表达了谢意。

如有抄袭，本人原因承担因此而造成的任何后果。

特此声明。

签名：刘帅

日期：2021年4月30日

程序引用说明

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 引用项 | 来源 | 相同代码行数 |
| 1 |  |  |  |
| 2 |  |  |  |
| 3 |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
| 小计 | | |  |

总代码行数\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_; 引用占比\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1、实验简介

本实验为双链表构造实验，利用头插法和尾插法构造双向链表，并实现了查找链表数据、对链表数据排序、获取链表长度信息、打印链表、删除链表、销毁链表等功能。

2、程序框架

1. 与构造相关的函数：

linklist() { front = new node; front->next = 0; }//空表

Linklist(int a[],int n)//利用构造函数构造链表（采用尾插法）

int headlinklist(int a[], int n);//头插法构造链表

2．与查找相关的函数：

void search(int searchnum);//查找链表数据

void listsort();//对链表数据进行搜索

int getlength();//获取链表长度信息

void quicksort(int\*arr,int l, int r);利用快速排序实现链表中数值的排序（复杂度为nlogn）

3、打印

void printlist();//打印链表

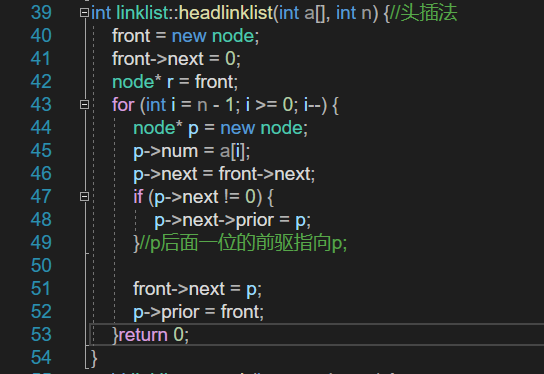
4、链表的删除

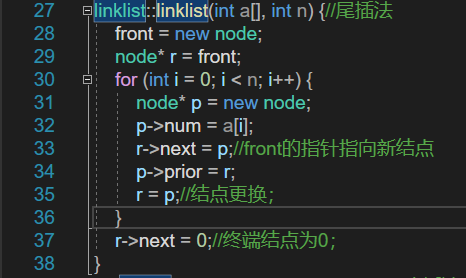
~linklist();//销毁链表

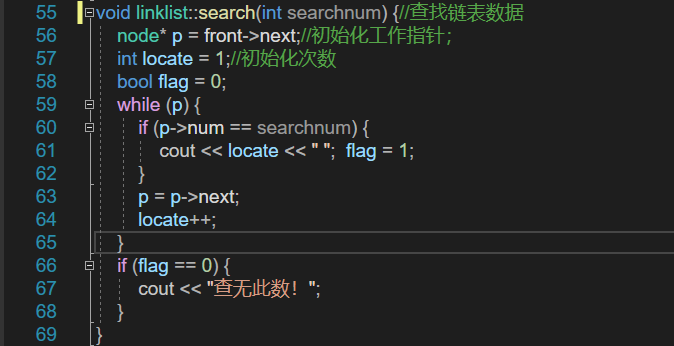
void Delete(int i);//删除链表节点；

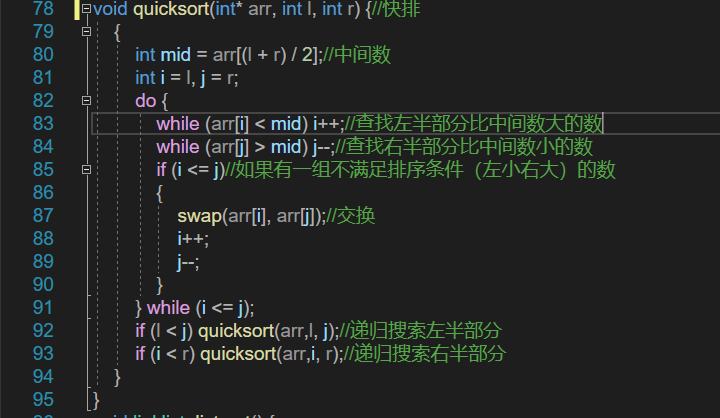
3、关键代码实现

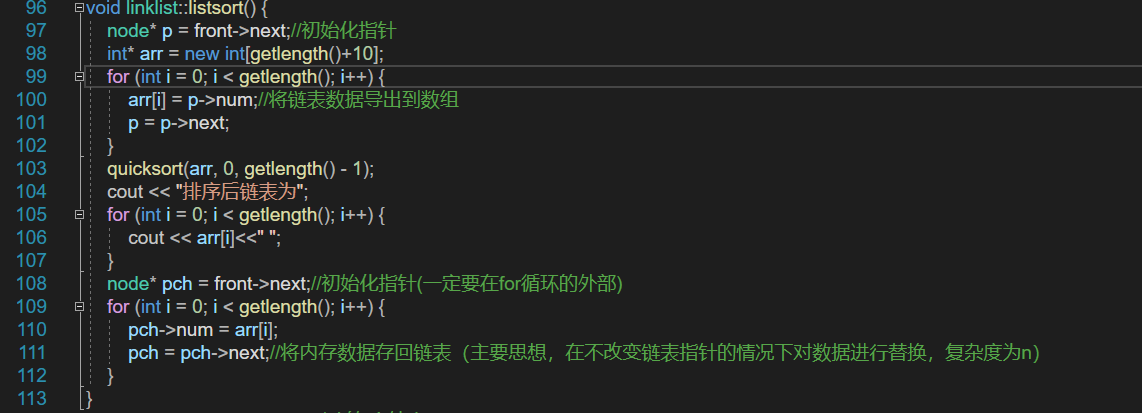
【哪些函数是你认为最能体现自己工作成果的函数，说明函数实现基本思想（可用文字或图表示），以及具体的实验步骤（用伪代码或带注释代码）】

3.1 int linklist::headlinklist(int a[], int n) //头插法

3.2 linklist::linklist(int a[], int n)//尾插法

3.3 void linklist::search(int searchnum) //查找链表数据

3.4 void quicksort(int\* arr, int l, int r) //快速排序

3.6 void linklist::listsort() //对链表数据进行搜索，主要思想为：先将链表数据导入到动态数组，对数组数据进行排序，然后再存回链表，存回链表操作复杂度为n。

4、不足

1、删除节点处仍有欠缺，对溢出情况的考虑不周。

2、对题意：“使用头插法、尾插法两种方法”理解有点不清（我理解的是想考察头插和尾插的写法，故将，链表的构造过程用尾插法写，并撰写了头插法的相关函数）

5、心得体会

由于本实验实际编程时间为四月初，因期中考试的缘故一直拖到了四月底才写完实验报告…在此过程中随着实验内容的深入逐渐对链表有了更深的理解，再回头看第一个实验则稍显幼稚，对于有些环节（如删除部分的溢出问题）的理解还缺成熟。

数据结构与算法

实验报告

实验题目：线性表题目5应用实验（高精度加减）

姓 名：刘帅

学 号：2020212267

日 期：2021年4月30日

自我评分：【 A+ 】

自我评分说明：A+，A，B+，B，B-，C，D，分别对应分数95、90、85、80、75、70、60

诚信声明

本人郑重承诺：本实验程序和实验报告均是本人独立学习和工作所获得的成果。尽我所知，实验报告中除特别标注的地方外，不包含其他同学已经发表或撰写过的成果；实验程序中对代码工作的任何帮助者所作的贡献均做了明确的说明，并表达了谢意。

如有抄袭，本人原因承担因此而造成的任何后果。

特此声明。

签名：刘帅

日期：2021年4月30日

程序引用说明

序号 引用项 来源 相同代码行数

1 无

小计

总代码行数205; 引用占比\_\_\_\_\_\_0%\_\_\_\_\_\_\_\_\_\_

1、实验简介

本实验为线性表的扩展实验，通过构造双链表实现高精度的加减法，并将结果存至新链表中。

2、程序框架

1. 与构造相关的函数：

largeint(int\*arr, int n);//利用尾插法建立数组

2．与操作有关函数：

void largeadd(largeint a, largeint b,largeint c);//高精度加法

void largeminus(largeint a, largeint b, largeint c);//高精度减法

void fill(largeint a, largeint b);//补全位数，前面补零（即加减法位数补齐操作）

void deletezero();//去除无效的零

int getnum();//获取位数

3、打印

void print();//打印结果

3、关键代码实现

【哪些函数是你认为最能体现自己工作成果的函数，说明函数实现基本思想（可用文字或图表示），以及具体的实验步骤（用伪代码或带注释代码）】

3.1 largeint::largeint(int\*arr,int n) {//利用尾插法建立数组

3.2 void largeint::fill(largeint a, largeint b)//在前方补足零（位数对齐）

先利用getnum函数判断两数位数多少，利用头插法的方法，在位数少的链表前补零。

3.3 void largeint::largeadd(largeint a, largeint b,largeint c) //高精度加法

值得注意（也是调试时间最长的）的是判断首位是否大于10的问题，需要建立新节点。

3.4 void largeint::largeminus(largeint a, largeint b, largeint c)//高精度减法

3.5 int largeint::getnum()//获取链表位数

3.6 void largeint::deletezero() //去除位首的零

4、不足

1.感觉讨论部分仍可以简化，代码的美观性仍有不足。

2.对移指针的操作有些复杂，应在最初构造链表的时候选择循环链表，时间复杂度会下降很多，属于决策失误。

5、心得体会

本实验让我深刻理解了链表的数据结构及各种线性表的优势之处，更让我在对位加减的实验过程中对存储结构有了深刻认识。