南开大学

Nankai University

南　开　大　学

网 络 空 间 安 全 学 院

数据结构实验报告

第二次实验作业

罗功成 1910487

年级：2019 级

专业：信息安全

指导教师：王玮

2021 年 10 月 26 日

摘要

对单链表进行输入若干多个含相同数值的数据，删除重复数字后进行拆分和排序

关键字：链表的实现、排序、去重和拆分，动态内存分配，复杂度分析

目录

一、 实验流程 1

（一） 实验内容和目的 1

（二） 实验原理和思路 1

（三） 实验核心代码 1

（四） 实验结果与分析 5

二、 实验探究问题解析 6

三、 总结和收获 6

一、 实验流程

(一) 实验内容和目的

在一个含有头节点的单链表 A 中，自行输入必须含有重复数据值的元素值。

要求：

1. 对 A 中出现数值重复的结点，仅保留第一次出现的结点
2. 对新得到的链表 A，根据数值的奇偶性进行拆分，拆分得到有序链表 B 和 C。
3. 链表 B 是数值递增的奇数链表，链表 C 是数值递增的偶数链表。
4. 注意实验顺序不能颠倒，在每一小问完成后输出对应结果，并对算法进行复杂度分析。

(二) 实验原理和思路

1. 链表的创建时，注意初始化数据个数不确定，需要加一次输入确认操作。
2. n 不确定，需要用动态内存空间分配的方式，用数组 a 存放初始数据。
3. 从小到大排序，考虑在链表中以指针方式实现冒泡排序。
4. 删除重复元素，仅保留第一个：考虑用两重循环的方式，第一个进行遍历，第二个选定在第一个后面进行遍历，凡是重复元素，直接对应删除即可。
5. 同理，在声明删除函数的时候引入一个计数量，当第一次遇到后先直接掠过，等到下一次遇到再删除。
6. 由于顺序都是递增的，因此考虑可以先对其排序，再去重，并且与两个拆分开后排序的方式相比，还可以少一次排序的过程。
7. 注意头节点没有数值！从下一个节点开始才是第一个数据。
8. 注意按步走，第一步只允许删除，不允许排序！所以要先删除，在第二步中再实现排序！

(三) 实验核心代码

链表创建部分

```
struct node {
        int data;
        node* next;
};

//创建链表
node* create(int array[]) {
        node* p, * head, * pre;//pre为前驱结点
        head = new node;
        head->next = NULL;
        pre = head;
        int n;
```

```
13          cin >> n;//创建可变大小的链表，但输入时需要注意这里还有一次输入，应与
                下面保持一致!
14          for (int i = 0; i < n; ++i) {
15                  p = new node;
16                  p->data = array[i];
17                  p->next = NULL;
18                  pre->next = p;
19                  pre = p;
20          }
21
22          return head;
23  }
24
25
26  // 插入
27  // 将x插入到以head为头节点的pos位置上
28  void insert(node* head, int pos, int x) {
29          node* p = head;
30          for (int i = 0; i < pos - 1; i++) {//比如要插到第三个位置，从0开始的
                话就是下标为2，前一个下标为1
31                  p = p->next;// 1 < 3-1
32          }
33          node* q = new node;
34          q->data = x;
35          q->next = p->next;
36          p->next = q;
37  }
38
39  //删除 ，重复值删除，保证只剩下第一次出现的
40  //删除所有值为x的数
41  void del(node* head, int x) {
42          node* p = head->next;
43          node* pre = head;
44          int count[1000];
45          for (int i = 0; i < 1000; i++)
46          {
47                  count[i] = 0;
48          }
49          while (p != NULL) {
50                  if  (p->data == x) {
51                          count[x]++;
52                          int m = x;
53                          if (count[x] == 1)
54                          {
55                                  pre = p;
56                                  p = p->next;//第一个不删
57                          }
58                          if (count[x] > 1)
```

```
                        {
                                pre->next = p->next;
                                delete(p);
                                p = pre->next;//到下一个开始再删
                        }
                }
                else {
                        pre = p;
                        p = p->next;
                }
        }
}

//查找
//查找链表中有几个x，返回count值
int search(node* head, int x) {
        int count = 0;
        node* p = head->next;
        while (p != NULL) {
                if (p->data == x) {
                        count++;
                }
                p = p->next;
        }
        return count;
}

//排序
//冒泡排序
void sort(node* head) {
        for (node* temp = head->next; temp->next != NULL; temp = temp->next)
            {
                for (node* p = head->next; p->next != NULL; p = p->next) {
                        if (p->data > p->next->data) {
                                int t = p->data;
                                p->data = p->next->data;
                                p->next->data = t;
                        }
                }
        }
}
```

创建，排序，去重

```
int n;
        cout << "请输入结点个数" << endl;
        cin >> n;
        int* a = new int[n];
        int* b = new int[n];
```

```
        int* c = new int[n];

        for (int i = 0; i < n; i++)
        {
                cin >> a[i];
        }
        cout << "注意，这里在定义L还有2个数目确认，需要和第一次输入值n一致！"
             << endl;
        node* L = create(a);//1.生成了初始链表A,以及后面的B,C链表
        node* L1 = create(a);
        int count = 0;
        //2.对链表先去重

        for (int i = 0; i < n; i++)
        {
                for (int j = i + 1; j < n; j++)//从i后开始遍历，遇到和前面相
                    同的元素直接删除。
                {
                        if (a[j] == a[i])
                        {
                                del(L, a[j]);
                                del(L1, a[j]);

                                count++;
                        }
                }

        }//完成了（1）
```

拆分给两个奇偶不同的升序链表

```
//(2)再排序
        cout << "删除重复节点链表A为：";
        L1 = L1->next;  //头节点L是没有数据域的，下个结点才有
        while (L1 != NULL) {
                printf("%d ", L1->data);
                L1 = L1->next;
        }

        sort(L);

        L = L->next;//头节点的下一个节点才有值

        //拆分链表：
        int count1 = 0; int count2 = 0;
        for (int i = 0; i < (n - count); i++)
        {
                if ((L->data) % 2 == 1)
                {
```

```
                        b[count1] = L->data;

                        L = L->next;

                        count1++;
                }
                else
                {
                        c[count2] = L->data;

                        L = L->next;
                        count2++;
                }
        }

        cout << "链表B为升序奇数链表" << endl;
        for (int i = 0; i < count1; i++)
                cout << b[i] << ' ';
        cout << endl;

        cout << "链表C为升序偶数链表" << endl;
        for (int i = 0; i < count2; i++)
                cout << c[i] << ' ';

        return 0;
}
                }
```

（四） 实验结果与分析

这里初始数据按照案例数据进行的，其中含有多个重复数字，并且没有连续挨着。

第一步初始数据排序去重为未排序状态，10,2，0，1，1，55，5，9，9，10，13，13，3，19.

变为 10 2 0 1 55 5 9 13 3 19

第二步拆分奇偶不同的升序链表:

B:1 3 5 9 13 19 55

C:0 2 10

```
请输入结点个数
14
10
2
0
1
1
55
5
9
9
10
13
13
3
19
注意，这里在定义L还有2个数目确认，需要和第一次输入值n一致！
14
14
删除重复节点链表A为：10 2 0 1 55 5 9 13 3 19
链表B为升序奇数链表
1 3 5 9 13 19 55
链表C为升序偶数链表
0 2 10
D:\LGC信息与基本资料\NKU南开大学学习资源库\3.大三（津南）：信息安全\大三上\数据结构\上机作业\数据结构\Debug\数据结构.exe
 (进程 58004)已退出，代码为 0。
按任意键关闭此窗口. . .
```

图 1: 先删除，再拆分排序

可见，以上代码可以满足本次实验的各项要求。

二、 实验探究问题解析

时间复杂度：O（n2）

主要是冒泡排序进行比较的过程，以及要删除对应数据的比对过程，均为 O（n2）

空间复杂度：O(n)

用到了 3 个 int 类型，长度为可变大小的 int 类型数组

SP=n*(addr(a)+sizeof(int)*3+addr(b)+addr(c)).

三、 总结和收获

1. 对于冒泡排序，动态内存分配，链表的创建、插入、删除、排序，类和对象等与本实验相关的代码进行了编程方面的复习。加深了自己对于链表这类数据结构算法实现的理解和认识。
2. 对课程学习中的时间复杂度和空间复杂度、实例特征、可变部分大小、关键操作的确认、渐进记法等和本实验相关的课程知识进行了复习和巩固。
3. 在实验过程中,在实现创建含有头节点的单链表的过程中借鉴了 https://zhuanlan.zhihu.com/p/14371834?from_voters_page=true 的部分内容，在学习和理解的基础上进行了完善和针对性的改进，在这里特别向该篇作者致谢。