

数据结构

实验二

单链表实验

2018年10月12日



实验二 单链表实验

深圳大学 蔡茂海

一、实验目的

- 掌握单链表的基本原理
- 掌握单链表的存储结构
- 掌握单链表的创建、查找、插入、删除的实现方法



实验二 单链表实验

深圳大学 基础部

二、实验要求

- 熟悉C++语言编程
- 熟练使用C++语言实现单链表的创建、插入、删除和查找



实验二 单链表实验

陈明 吴智 蔡茂海

三、实验内容

1、问题描述

- 给出初始数据
- 实现单链表的定义、创建、查找、插入、删除

实验二 单链表实验

三、实验内容

2、算法

- 单链表结点的存储结构包含两部分：数据、下一结点指针
- 单链表的查找：给出位置 i 结点的指针
- 单链表的插入：创建新的结点，在位置 $i-1$ 和位置 i 之间插入新的结点，修改相关指针指向
- 单链表的创建：依次为输入的数据分配结点，并按序链接起来
- 单链表的删除：修改位置 $i-1$ 结点的下一结点指针指向 $i+1$ 结点，删除位置 i 结点，释放被删除结点



实验二 单链表实验

陈时龙 蔡茂德

三、实验内容

3、输入

- 每个样本分4行:

- 第一行: 第一个数字 n 表示样本数目, 其后跟 n 个样本
- 第二行: 查找数据的位置
- 第三行: 插入新数据位置、新数据
- 第四行: 删除数据的位置



实验二 单链表实验

陈明 王智 蔡茂海

三、实验内容

4、输入样本

5 2 4 3 5 7

4

2 9

3



实验二 单链表实验

陈明 王 磊 编

三、实验内容

5、输出

- 第一行：单链表创建后，数据数组按顺序排列的全部数据
- 第二行：按指定位置给出数据
- 第三行：插入新数据后，数据数组按顺序排列的全部数据
- 第四行：删除一个数据，数据数组按顺序排列的全部数据



实验二 单链表实验

深圳大学 基础课

三、实验内容

6、输出样本

2 4 3 5 7

5

2 9 4 3 5 7

2 9 3 5 7



实验二 单链表实验

深圳大学 基础课

四、实验步骤

- 1、单链表的定义
- 2、单链表的查找
- 3、单链表的插入
- 4、单链表的创建
- 5、单链表的删除

实验二 单链表实验

深圳大学 蔡茂海

四、实验步骤

1、单链表的定义

- 定义一个单链表结点，其存储结构包含两部分：数据、下一结点指针

```
struct LNode {  
    int    data;           // 数据域  
    LNode *next;          // 后继指针  
};  
  
LNode *LinkList;
```

实验二 单链表实验

陈明 王宇 蔡茂 编

四、实验步骤

2、单链表的查找

- 输出指定位置(i)的数据结点指针
- 如果指定位置(i)超出范围，返回NULL
- 如果为空表，返回NULL

```
LNode *SearchLinkList(LNode *L, int i)
{
    // L为单链表的指针, e为返回值
    int j;
    LNode *p;
    p = L; j = 1;
    while (p && j < i) {p = p->next; j++;}
    if (!p || j > i) return (NULL);
    return (p);
} // SearchLinkList
```

// p指向第一个结点
// 顺指针查找
// 第i元素不存在
// 返回第i元素指针

实验二 单链表实验

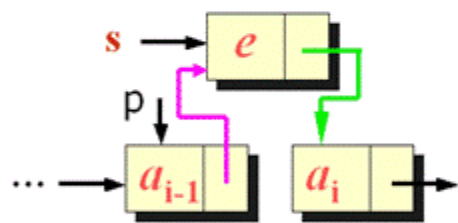
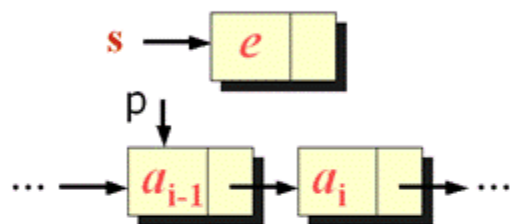
陈时天等 基础篇

四、实验步骤

3、单链表的插入

- 将新数据(e)插入到指定位置(i)上
- 如果指定位置(i)超出范围, 返回ERROR

```
int InsertLinkList(LNode *L, int e, int i)
{
    // 在单链表L中第i个位置插入元素e
    LNode *p, *s;
    p = SearchLinkList(L, i-1); if (!p) return(ERROR);
    s = new LNode;                // 生成新结点
    s->data = e;                  // 节点插入L中
    return(CORRECT);
} // InsertLinkList
```



实验二 单链表实验

四、实验步骤

4、单链表创建

- 将输入的关键字依次放入单链表中：

```
LNode *CreateLinkList(int *r, int n)
{
    int j;    LNode *L, *s;
    if (n <= 0) return(NULL);
    s = new LNode;
    s->data = r[1];
    s->next = NULL;           // 创建第1个结点
    L = s;                   // 单链表头指针指向第1个结点
    for (j=2; j<=n; j++) InsertLinkList(L, r[j], j);
    return(L);
}
```

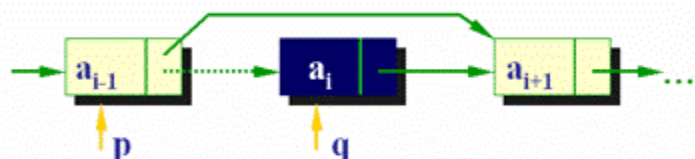
实验二 单链表实验

陈明 王智 蔡茂 编

四、实验步骤

5、单链表的删除

- 将指定位置(i)的数据元素删除
- 如果指定位置(i)超出范围, 返回ERROR
- 如果为空表, 返回ERROR



```
int DeleteLinkList(LNode *L, int i)
{    // 在单链表L中, 删除第i个位置的元素
    int e; LNode *p, *q;
    p = SearchLinkList(L, i-1); if (!p) return(ERROR);
    q = p->next; ..... // 删除i结点
    e = q->data; delete(q); // 取值并释放结点
    return(e);
} // DeleteLinkList
```

实验二 单链表实验

陈明 王 磊 等

四、实验步骤

6、结果输出

- 按要求的格式，输出结果

```
int ShowLinkList(LNode *L)           // 输出单链表结果，用于测试
{
    LNode *p;
    if (!L) return(ERROR);
    p = L;
    while (p->next) {
        cout << p->data <<" ";    //逐个显示数据元素
        p = p->next;
    }
    cout << p->data << endl;       //显示最后一个数据元素，并换行
    return(CORRECT);
}
```


实验二 单链表实验

陈明 王智 蔡茂强

四、实验步骤

7、主程序

```
int main(int argc, char* argv[])
{
    int r[100], i, SampleNo, SearchPos, NewPos, NewItem, DelPos; LNode *p;

    cin >> SampleNum; //输入样本数目
    for (i=1; i<=SampleNum; i++) cin >> r[i]; //输入样本数据
    LinkList=CreateLinkList(r, SampleNum); ShowLinkList(LinkList); //生成单链表, 并显示
    cin >> SearchPos; //输入查找位置
    p = SearchLinkList(LinkList, SearchPos);
    cout << p->data << endl; //输出查找的数据元素值
    cin >> NewPos; cin >> NewItem; //输入1个新数据位置和值
    InsertLinkList(LinkList, NewItem, NewPos); //新数据插入单链表中
    ShowLinkList(LinkList); //显示单链表
    cin >> DelPos; //输入删除元素位置
    DeleteLinkList(LinkList, DelPos); ShowLinkList(LinkList); //删除数据元素, 并显示
    return 0;
}
```



实验二 单链表实验

陈时龙 蔡茂海

四、参考材料

- 严蔚敏、吴伟民，《数据结构(C语言版)》，清华大学出版社，2015. 05