**实验五 哈希表**

PB17050941 李喆昊

1. 问题描述

题目：哈希表

1.  输入关键字序列；

2.  用除留余数法构建哈希函数，用线性探测法解决冲突，构建哈希表HT1；

3.  用除留余数法构建哈希函数，用拉链法解决冲突，构建哈希表HT2；

4.  分别对HT1和HT2计算在等概率情况下查找成功和查找失败的ASL；

5.  分别在HT1和HT2中查找给定的关键字，给出比较次数。

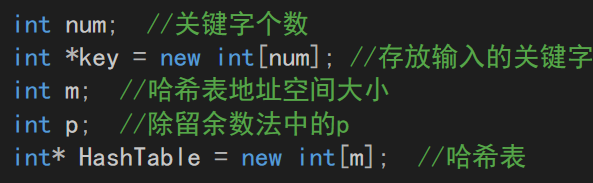
输入：测试文件中包括：关键字个数n，关键字key（这里我们认为关键字key就是哈希表中元素对应的哈希函数值），和除留余数法中的p。

输出：要求按顺序输入关键字后，输出哈希表的相关信息（表的结构、成功查找次数、失败查找次数、平均成功查找长度、平均失败查找长度）。

1. 算法描述
2. 数据结构描述
3. HT1

采用线性表作为存储结构，逻辑结构为线性结构。

主要变量：



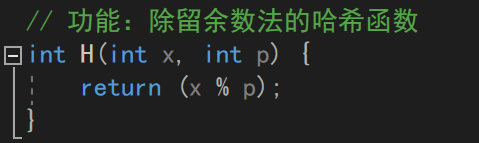
1. HT2

采用线性链表作为存储结构，逻辑结构为线性结构。

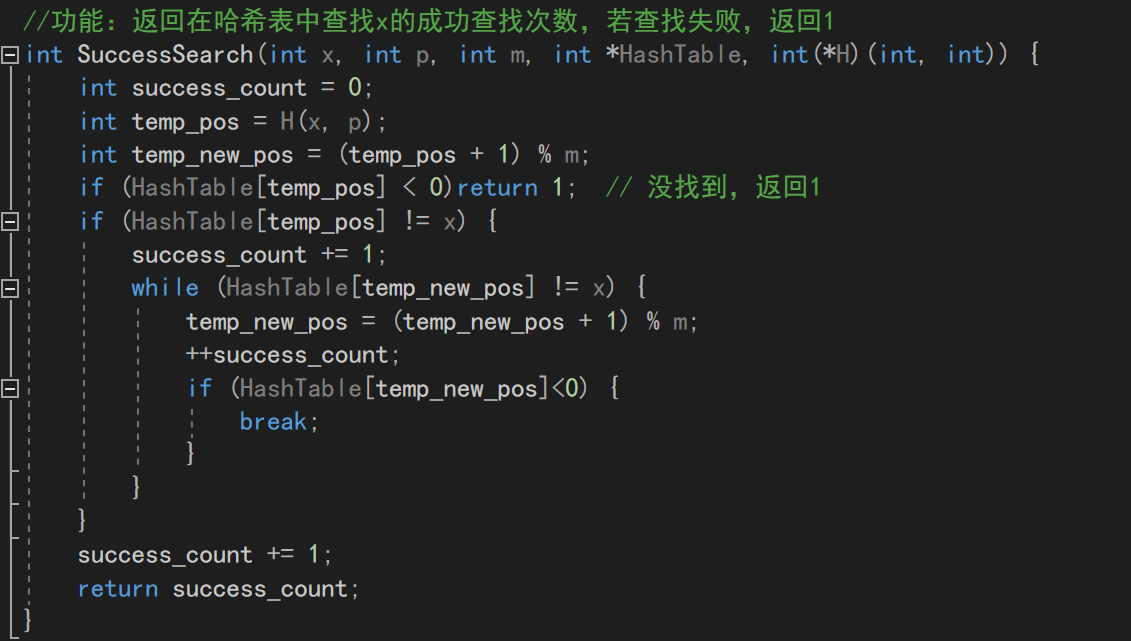
主要变量：



1. 程序结构描述
2. HT1



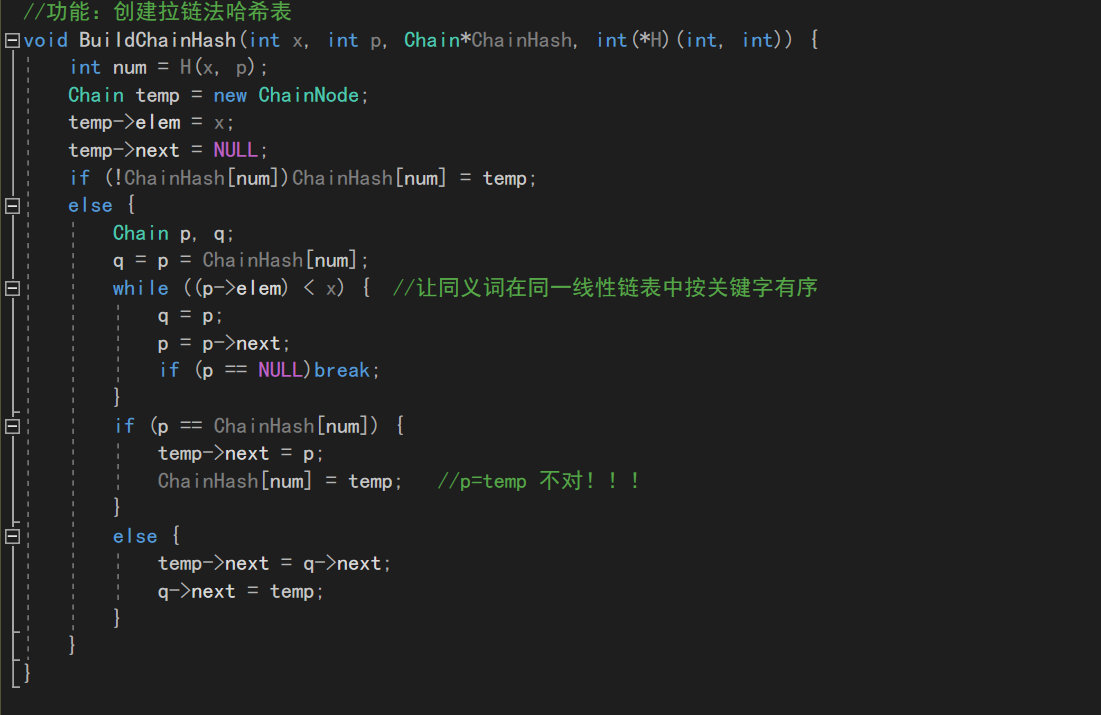
接口：x为待查找的关键字，p为除留余数法中的除数p



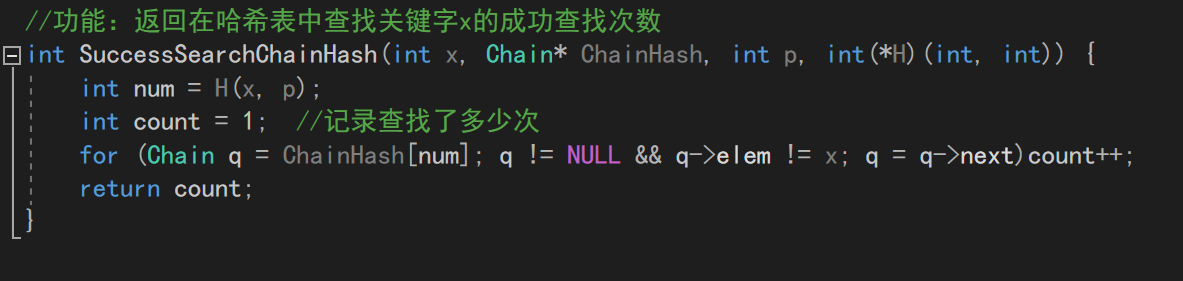
接口：x，p同上，m为哈希表的大小，HashTable为指向哈希表首地址的指针，H为哈希函数指针。

1. HT2

哈希函数同（1)。



接口：x为待查找的关键字，p为除留余数法中的除数p，ChainHash为指向拉链法哈希表表头数组的首地址的指针，H为哈希函数指针。



接口同上。

三．调试分析

（1）测试数据使用的是助教提供的五组测试数据。

（2）程序调试中遇到的问题：

① 拉链法创建哈希表时出现了创建错误的现象。

解决方法：仔细检查BuildChainHash函数定义，发现是“让关键字在每个链表内按从大到小有序排列”一步算法有瑕疵。

②测试时发现平均失败查找次数与标准答案不同。

解决方法：检查失败查找次数统计步骤，最后发现是哈希表的地址空间大小设置的与答案不同。

四．算法时空分析

1. HT1

①创建哈希表数组：因为采用线性探测法，时间复杂度为 O(n)，空间复杂度也为O(n)

②查找：时间O(n)，空间O(1)（只需要一些固定大小的辅助空间）

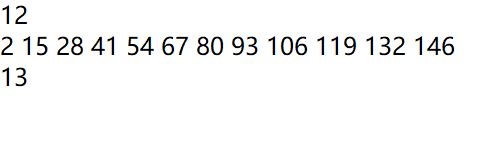
1. HT2

①创建哈希表：因为采用拉链法，时间复杂度为 O(n)，空间复杂度也为O(n)

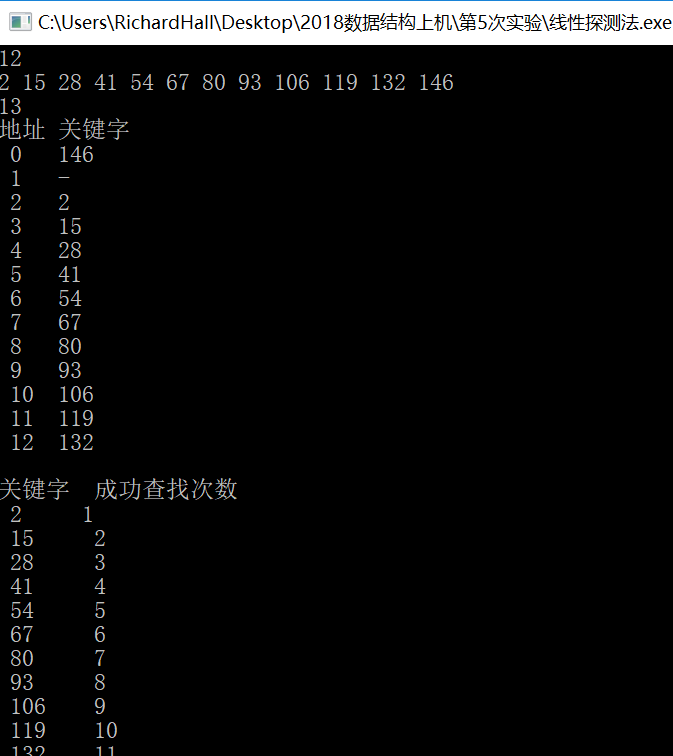
②查找：时间O(n)，空间O(1)（只需要一些固定大小的辅助空间）

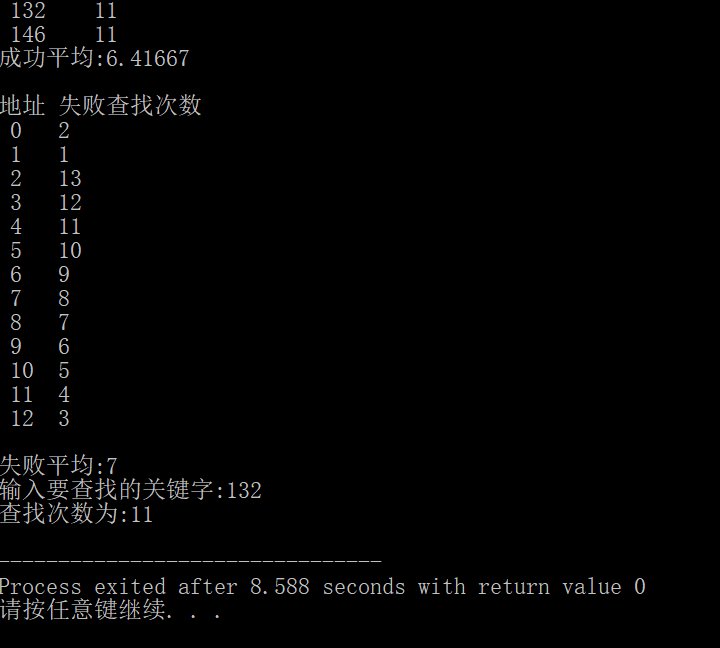
五．测试结果及分析

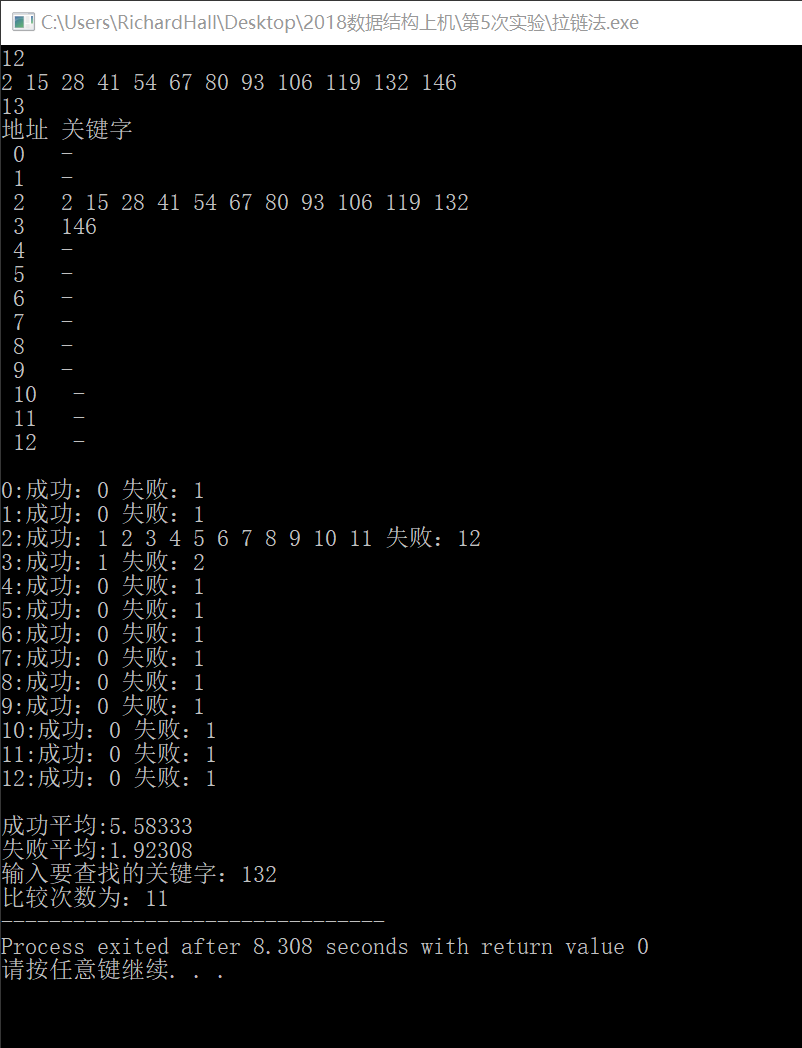
例：测试数据为



1. HT1





1. HT2

结果分析：我们可以清晰地看到，线性探测法和拉链法得到的哈希表形态非常不同。

正因为如此，拉链法的平均查找长度一般小于线性探测法发平均查找长度。

六．实验体会和收获

1. 增进了对哈希表的理解，学习了哈希表中处理冲突的两种方式：线性探测法和拉链法，也体会到了哈希表查找相比于传统查找方式的优势。
2. 锻炼了使用编程处理问题的能力。