**《数据结构》上机报告**

**2018 年 12 月 13 日**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **姓名：** | **赵得泽** | **学号：** | **1753642** | **班级：** | **电子2班** | **得分：** |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **实验题目** | | **哈希表（Hash Table）** | | |
| **问题描述** | | 哈希表（hash table,散列表）是一种用于以常数平均时间执行插入、删除和查找的查找表，其基本思想是：找到一个从关键字到查找表的地址的映射h（称为散列函数），将关键字key的元素存到h(key)所指示的存储单元中。当两个不相等的关键字被散列到同一个值时称为冲突，产生冲突的两个（或多个）关键字称为同义词，冲突处理的方法主要有：开放定址法，再哈希法，链地址法。 | | |
| **基本要求** | | 1. 本题针对字符串设计哈希函数。假定有一个班级的人名名单，用汉语拼音（英文字母）表示。   要求：   1. 首先把人名转换成整数，采用函数h(key)=key[n-1]37^(n-1)+key[n-2]37^(n-2)+...+key[1]\*37+key[0]，其中key[i]表示人名从右往左的第i个字母(i从0计数)在字母表中的序号，例如字母a和A的序号都是1。 2. 采取除留余数法（模是M）将整数映射到长度为p的散列表中，h(key)=h(key)%M； 3. 采用线性探测法解决冲突；   **注意：计算h(key)时会发生溢出，需要先取模再计算。** | | |
| **已完成基本内容（序号）：** | 1 | |
| **选做要求** | |  | | |
| **已完成选做内容（序号）** | |  |
| **数据结构**  **设计** | | **typedef char\* ElemType;**  **typedef struct**  **{**  **ElemType \*elem;**  **}HashTable;**  **int letter[MAX\_SIZE];**  **int \_count[MAX\_SIZE];**  **本实验的数据结构主要是线性结构中的顺序表，其中的数据是一个二维数组，在记录地址的同时进行数据的存储，还有一个\_count[]数组对每个数据查找到地址需要的次数进行了统计，letter[]数组则是对每个字符串按输入顺序进行了地址的记录。** | | |
| **功能(函数)说明** | | **/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***  **函数功能：建立哈希表**  **函数说明：其核心做法就是对输入的数据利用哈希函数映射到相应的地址上，**  **在此采用了除留余数法（模是M）将整数映射到长度为p的散列表中，具体方**  **法见下列函数注释。**  **\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/**  **void Create\_HT(HashTable &HT, ElemType key, int r)**  **{**  **int k = 0, l;**  **l = 37 % mod;//防止数据过大溢出**  **for (int j = 0; j<strlen(key); j++)**  **{**  **key[j] = (key[j] >= 'a'&&key[j] <= 'z') ? key[j] - 'a' + 1 : key[j] - 'A' + 1;//对大小写进行区分**  **k = k + (((int)key[j])\* pow(l, strlen(key) - j-1));//对输入的字符串进行哈希函数转换**  **}**  **k = k % mod;//再次取模找到映射地址**  **int q = 1;**  **while (1)**  **{**  **if (strlen(HT.elem[k]) == 0)//如果哈希表中相应地址中的数据域为空，则将其用字符串填充**  **{**  **strcpy(HT.elem[k], key);**  **letter[r] = k;//字符串对应的地址**  **\_count[r] = q;//字符串找到地址时候查找的次数**  **break;**  **}**  **else//若该地址处已经有了数据，即有冲突，则利用线性探测法解决冲突。**  **{**  **k++;**  **q++;**  **k = k % mod;**  **}**  **}**  **}** | | |
| **开发环境** | | Win10 ,vs2017,C++高级程序语言设计 | | |
| **调试分析** | |  | | |
| **心得体会** | 本实验主要是哈希表的应用，即找到一个从关键字到查找表的地址的映射h（称为散列函数），将关键字key的元素存到h(key)所指示的存储单元中。在本题目中主要就是将字符串存入elem二维数组中，然后利用题目已知的散列函数计算出每个字符串的地址，然后将key字符串存到elem[]中，在此过程中可能会遇到冲突，即两个字符串的地址相同，那么就还要对冲突进行解决，在此用到的是**线性探测法**解决冲突，当然解决冲突的方法还有**开放定址法，再哈希法，链地址法。**  通过对哈希表的一个简单应用，我们可以看到哈希表的查找的主要步骤就是**利用哈希函数找到对应数据的映射地址**，以及遇到冲突时的解决方法的选择。 | | | |