



数据结构与算法实验

实验八 散列表

彭振辉

中山大学人工智能学院

2023年秋季学期



实验任务1

- 实现如下线性探测法的查找函数

Position **Find**(HashTable H, ElementType key)

HashTable 定义

```
#define MAXTABLESIZE 100000 /* 允许开辟的最大散列表长度 */
typedef int ElementType; /* 关键词类型用整型 */
typedef int Index; /* 散列地址类型 */
typedef Index Position; /* 数据所在位置与散列地址是同一类型 */
/* 散列单元状态类型，分别对应：有合法元素、空单元、有已删除元素 */
typedef enum { Legitimate, Empty, Deleted } EntryType;
typedef struct HashEntry Cell; /* 散列表单元类型 */
struct HashEntry{
    ElementType Data; /* 存放元素 */
    EntryType Info; /* 单元状态 */
};
typedef struct TblNode *HashTable; /* 散列表类型 */
struct TblNode { /* 散列表结点定义 */
    int TableSize; /* 表的最大长度 */
    Cell *Cells; /* 存放散列单元数据的数组 */
};
```

实验任务1



示例:

Input1:

11

11 88 21 -1 -1 5 16 7 6 38 10

38

Output1:

38 is at position 9.

Input2:

11

11 88 21 -1 -1 5 16 7 6 38 10

41

Output2:

41 is not found. Position 3 is returned.

注意:

- 函数Find应根据裁判定义的散列函数 Hash(Key, H->TableSize)从散列表H中查到Key的位置并返回。如果Key不存在, 则返回线性探测法找到的第一个空单元的位置; 若没有空单元, 则返回ERROR。
- -1表示该位置为空

实验任务2



- 将给定的无重复正整数序列插入一个散列表，输出每个输入的数字在表中的位置。所用的散列函数是 $H(\text{key}) = \text{key} \% \text{TSize}$ ，其中 TSize 是散列表的表长。要求用平方探测法 (只增不减，即 $H(\text{Key}) + i^2$) 解决冲突。注意散列表的表长最好是个素数。若输入给定的表长不是素数，你必须将表长重新定义为大于给定表长的最小素数

示例：

Input1:

4 4

10 6 4 15

Output1:

0 1 4 -

- 输入：首先第一行给出两个正整数 Msize (≤ 10000) 和 N ($\leq \text{Msize}$)，分别对应输入的表长和输入数字的个数。随后第二行给出 N 个不重复的正整数，数字间以空格分隔
- 输出：在一行中按照输入的顺序给出每个数字在散列表中的位置(下标从 0 开始)。如果某个数字无法插入，就在其位置上输出 “-”。输出间以 1 个空格分隔，行首尾不得有多余空格

实验任务3



- 试实现如下分离链接法的删除操作函数

```
bool Delete( HashTable H, ElementType key )
```

HashTable 定义:

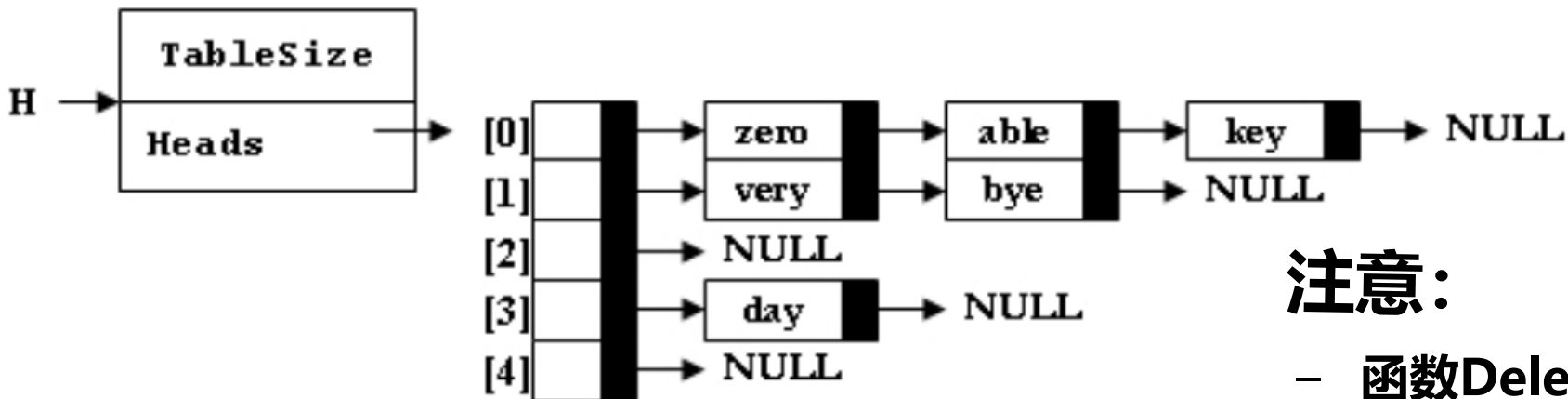
```
typedef struct LNode *PtrToLNode;
struct LNode {
    ElementType Data;
    PtrToLNode Next;
};
typedef PtrToLNode Position;
typedef PtrToLNode List;

typedef struct TblNode *HashTable; /* 散列表类型 */
struct TblNode { /* 散列表结点定义 */
    int TableSize; /* 表的最大长度 */
    List Heads; /* 指向链表头结点的数组 */
};
```

实验任务3



示例:



Input:

able

Output:

able is deleted from list Heads[0]

注意:

- 函数Delete应根据裁判定义的散列函数Hash(Key, H->TableSize)从散列表H中查到Key的位置并删除之, 然后输出一行文字: Key is deleted from list Heads[i], 其中Key是传入的被删除的关键词, i是Key所在的链表的编号; 最后返回true。如果Key不存在, 则返回false

实验任务4



给定一系列由大写英文字母组成的字符串关键字和素数 P ，用移位法定义的散列函数 $H(Key)$ 将关键字 Key 中的最后3个字符映射为整数，每个字符占5位；再用除留余数法将整数映射到长度为 P 的散列表中。例如将字符串 **AZDEG** 插入长度为1009的散列表中，我们首先将26个大写英文字母顺序映射到整数0~25；再通过移位将其映射为 $3 \times 32^2 + 4 \times 32 + 6 = 3206$ ；然后根据表长得到 $3206 \% 1009 = 179$ ，即是该字符串的散列映射位置。

发生冲突时请用平方探测法解决。

实验任务4



输入第一行首先给出两个正整数 N (≤ 500) 和 P ($\geq 2N$ 的最小素数)，分别为待插入的关键字总数、以及散列表的长度。第二行给出 N 个字符串关键字，每个长度不超过8位，其间以空格分隔。

示例输入：

4 11

HELLO ANNK ZOE LOLI

在一行内输出每个字符串关键字在散列表中的位置。数字间以空格分隔，但行末尾不得有多余空格。

示例输出：

3 10 4 0