



目录
CONTENTS

01

【哈希表的相关的定义】

02

【哈希函数的构建】



01

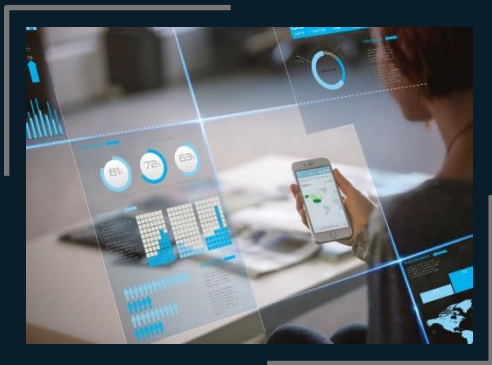
哈希表的相关定义

查找表的各种结构的共同特点：

- 记录在表中的位置和它的关键字之间不存在一个确定的关系；
- 查找的过程为给定值依次和关键字集合中各个关键字进行比较；
- 查找算法的效率取决于和给定值进行比较的关键字个数。

关键字和给定值进行比较的顺序不同。对于频繁使用的查找表，我们希望平均查找长度0，即理想情况是希望不经过任何比较一次存取就能找到要查找的记录。

一、哈希表的相关定义



若查找结构中存在关键字和 K 相等的记录，则必定在 $f(K)$ 的存储位置上，因此，不需要进行比较便可直接取得所查记录，这里，我们称这个对应关系 f 为**哈希函数**， $f(K)$ 为**哈希地址**，按这个思想建立的查找表叫做**哈希表**。

例2：对于如下 7 个关键字

{zj(zhejiang), qd(qingdao), sd(shandong),
ln(liaoning), yn(yunnan), hn(henan), bj(beijing)}

设 哈希函数 $f(\text{key}) = \lfloor (\text{Ord}(\text{第一个字母}) - \text{Ord}('A') + 1) / 2 \rfloor$

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|---|----|---|---|----|---|----|---|----|----|----|----|----|----|
| | bj | | | hn | | ln | | qd | sd | | | yn | zj |

问题：若添加关键字(bz) binzhou , 怎么办？

$f(\text{tianjin}) = (\text{Ord}(\text{第一个单词的首字母}) + \text{Ord}(\text{最后一个字母})) - 30 = 34 - 30 = 4$

| | | | | | | | | | | | | | |
|---|---|---|---|-----------|---|---|---|---|---|----|----|----|----|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
| | | | | tj | | | | | | | | | |

$f(\text{shanxi}) = f(\text{shanghai})$

总结：

- 哈希函数是一个映像，因此哈希函数的设定很灵活并不是唯一的，只要使得任何关键字通过哈希函数得到的值都落在表长允许的范围之内即可。
- 对不同的关键字可能得到同一个地址，比如之前例子里的beijing 和binzhou，也就是关键字key1不等于关键字key2，但是关键字key1和关键字key2的哈希地址是相同的。这种现象叫做冲突。

01

哈希表的构造方法



二、构造哈希函数的方法

对数字的关键字可有下列构造方法：

1. 直接定址法

4. 折叠法

2. 数字分析法

5. 除留余数法

3. 平方取中法

6. 随机数法

注：若关键字非数字，则需先对其进行数字化处理。

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/988070073057006044>