

2020/06/10_数据结构_第9课_哈希表

笔记本： 数据结构

创建时间： 2020/6/11 星期四 11:10

作者： ileemi

- [哈希表](#)
- [哈希表的构建](#)
- [哈希函数的构造](#)
- [处理冲突的方法](#)

哈希表

哈希表可以通过关键字直接找到数据的存储位置，不需要进行任何的比较，其查找的效率相较于前面所介绍的查找算法是更高的。

哈希表的构建

在初中的数学课本中学习过函数的相关知识，给定一个 x ，通过一个数学公式，只需要将 x 的值带入公式就可以求出一个新的值 y 。

哈希表的建立同函数类似，把函数中的 x 用查找记录时使用的关键字来代替，然后将关键字的值带入一个精心设计的公式中，就可以求出一个值，用这个值来表示记录存储的哈希地址。即：

数据的哈希地址 = f (关键字的值)

哈希地址只是表示在查找表中的存储位置，而不是实际的物理存储位置。 $f()$ 是一个函数，通过这个函数可以快速求出该关键字对应的的数据的哈希地址，称之为“哈希函数”。

例如，这里有一个电话簿（查找表），电话簿中有 4 个人的联系方式：

张三 13912345678

李四 15823457890

王五 13409872338

赵六 13805834722

假如想查找李四的电话号码，对于一般的查找方式最先想到的是从头遍历，——比较。而如果将电话簿构建一张哈希表，可以直接通过名字“李四”直接找到电话号码在表中的位置。

在构建哈希表时，最重要的是哈希函数的设计。例如设计电话簿案例中的哈希函数为：每个名字的姓的首字母的 ASCII 值即为对应的电话号码的存储位置。这时会发

现，张三和赵六两个关键字的姓的首字母都是 Z，最终求出的电话号码的存储位置相同，这种现象称为冲突。在设计哈希函数时，要尽量地避免冲突现象的发生。

对于哈希表而言，冲突只能尽可能地少，无法完全避免。

哈希函数的构造

常用的哈希函数的构造方法有 6 种：直接定址法、数字分析法、平方取中法、折叠法、除留余数法和随机数法。

如此多的构建哈希函数的方法，在选择的时候，需要根据实际的查找表的情况采取适当的方法。通常考虑的因素有以下几方面：

- 关键字的长度。如果长度不等，就选用随机数法。如果关键字位数较多，就选用折叠法或者数字分析法；反之如果位数较短，可以考虑平方取中法；
- 哈希表的大小。如果大小已知，可以选用除留余数法；
- 关键字的分布情况；
- 查找表的查找频率；
- 计算哈希函数所需的时间（包括硬件指令的因素）

处理冲突的方法

对于哈希表的建立，需要选取合适的哈希函数，但是对于无法避免的冲突，需要采取适当的措施去处理。

通常用的处理冲突的方法有以下几种：

- 开放定址法
- 再哈希法
当通过哈希函数求得的哈希地址同其他关键字产生冲突时，使用另一个哈希函数计算，直到冲突不再发生
- 链地址法
将所有产生冲突的关键字所对应的数据全部存储在同一个线性链表中
- 建立一个公共溢出区
建立两张表，一张为基本表，另一张为溢出表。基本表存储没有发生冲突的数据，当关键字由哈希函数生成的哈希地址产生冲突时，就将数据填入溢出表。

参考文章：[哈希表（散列表）及哈希表处理冲突的方法](#)