

3.1 哈希表介绍_物联网 / 嵌入式工程师 - 慕课网

“ 慕课网慕课教程 3.1 哈希表介绍涵盖海量编程基础技术教程，以图文图表的形式，把晦涩难懂的编程专业用语，以通俗易懂的方式呈现给用户。

1. 哈希表介绍

哈希表 (Hash table, 也叫散列表)，是根据关键码值 (Key value) 而直接进行访问的一种数据结构。也就是说，它通过把关键 码值映射到表中一个位置来访问记录，以加快查找的速度。这个映

射函数叫做散列函数，存放记录的数组叫做散列表。

记录的存储位置 = f(关键字)

这里的对应关系 f 称为散列函数，又称为哈希 (Hash 函数)，采用

散列技术将记录存储在一块连续的存储空间中，这块连续存储空间

称为散列表或哈希表 (Hash table)。

假设我们有 5 个数据。存放到一个数组中，数据如下：

```
int a[5] = {10,22,13,11,24}.
```

若是我们想要找到 13 这个元素，通过以下代码我们循环 3 个基

本上来说就可找到。

```
for(i = 0; i < 5; i++)
```

```
{
```

```
if(a[ i ] == 13)
```

```
break;
```

```
}
```

以上方法效率较低。

$$f(\text{key}) = \text{key} \% p$$

注：

key 我们需要存储的有效数据，

p 为了哈希表的长度。

f(key) 为通过哈希函数计算出来的新的数组的下标。

int a[5] = {10, 22, 13, 11, 24};

01234

1022131124

正常存储:

int hash[5];

01234

1011221324

哈希存储:

思考:若是我们想要查找我们的13这个元素的话,通过下标for循环查找的需要几次?

答案: 3次

哈希函数:

$h(k) = k \% 5;$

$h(10) = 0;$
 $h(22) = 2;$
 $h(13) = 3;$
 $h(11) = 1;$
 $h(24) = 4;$

注:若是我们的数组原来是一组特殊的数据,那么就会存在一定的冲突。

```
int a[7] = {10,22,13,11,24,7,14};

f(key) = key % 7;
f(10) = 3;
f(22) = 1;
f(13) = 6;
f(11) = 4;

f(24) = 3 ;
```

全文完

本文由 简悦 SimpRead 优化,用以提升阅读体验

使用了 全新的简悦词法分析引擎 beta, 点击查看详细说明

