

048 底层分析

一、字典核心底层原理

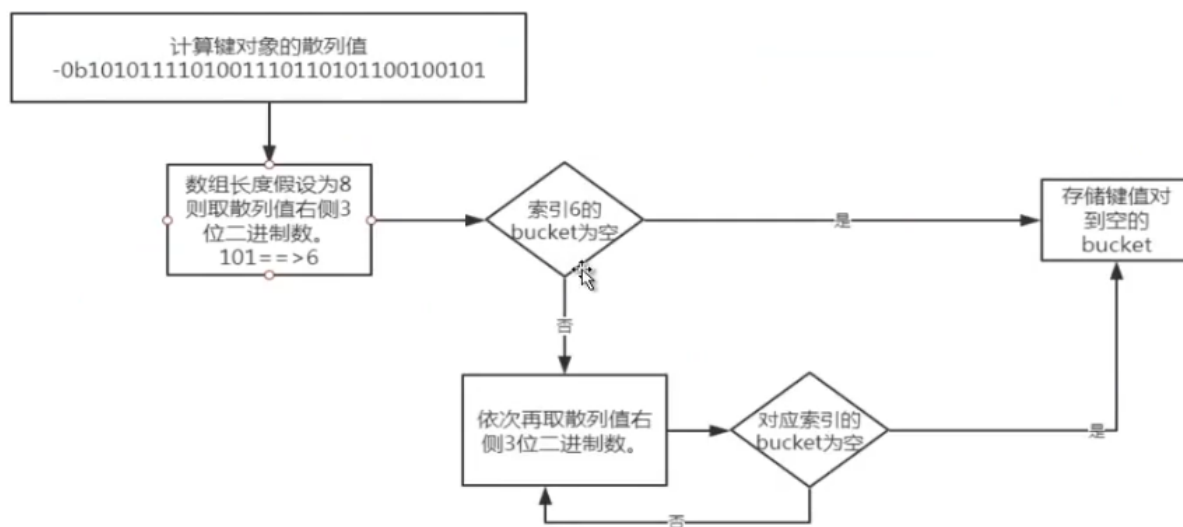
字典对象的核心是散列表。散列表是一个系数数组（总有空白元素的数组），数组的每一个单元叫做bucket。每一个bucket有两部分：一个是键对象的引用，一个是值对象的引用。

通过偏移量来指定bucket。

二、将一个键值对放进字典的底层过程

假设字典 a 对象创建完后，数组长度为 8：

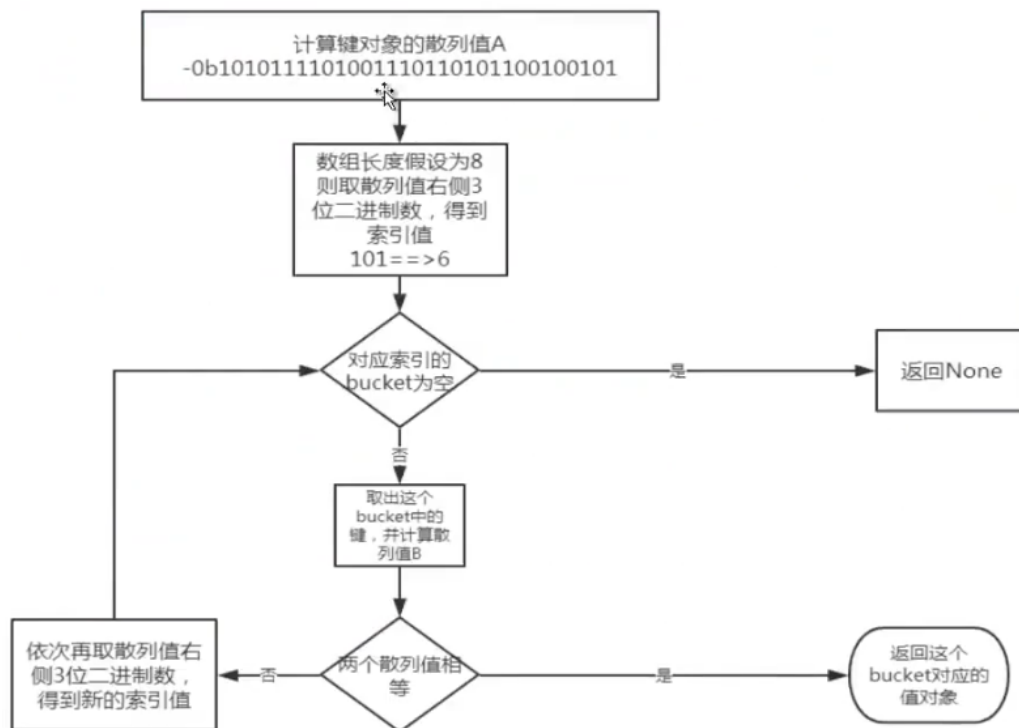
0	
1	
2	
3	
4	
5	
6	
7	



0	
1	
2	
3	
4	
5	
6	name gaoqi
7	

只是示意，真实存储的是
"name"对象的地址。
"gaoqi"字符串的地址

三、根据键查找键值对的底层过程



用法总结：

1. 键必须可散列

- (1) 数字、字符串、元组，都是可散列的。
- (2) 自定义对象需要支持下面三点：
 - ① 支持 hash() 函数
 - ② 支持通过 `__eq__()` 方法检测相等性。
 - ③ 若 `a==b` 为真，则 `hash(a)==hash(b)` 也为真。

2. 字典在内存中开销巨大，典型的空间换时间。

3. 键查询速度很快

4. 往字典里面添加新建可能导致扩容，导致散列表中键的次序变化。因此，不要在遍历字典的同时进行字典的修改。