

# 哈希表的原理

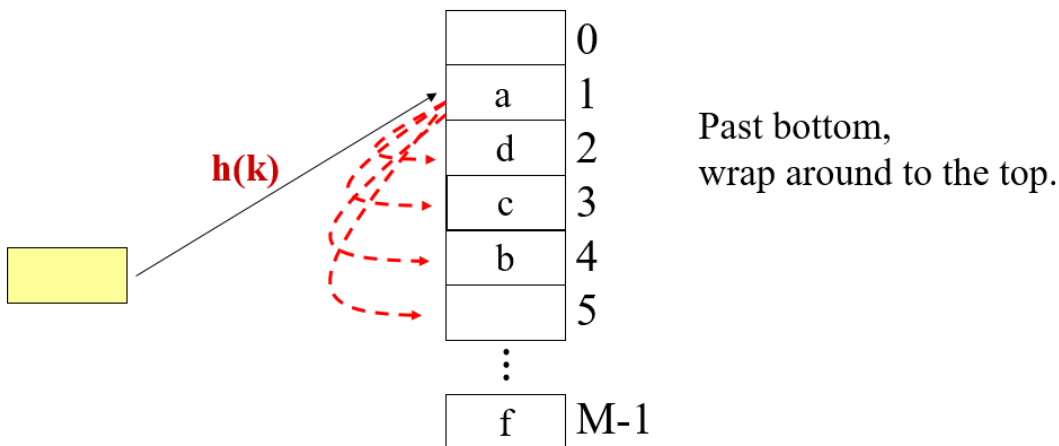
- 1、哈希表（Hash table，也叫散列表），是根据关键码值(Key value)而直接进行访问的数据结构。
- 2、散列函数：能使对一个数据序列的访问过程更加迅速有效，通过散列函数，数据元素将被更快地定位。
- 3、处理冲突：开放地址法与链地址法。

开放地址法有线性探查、二次再探测、双重哈希等。

线性探查：

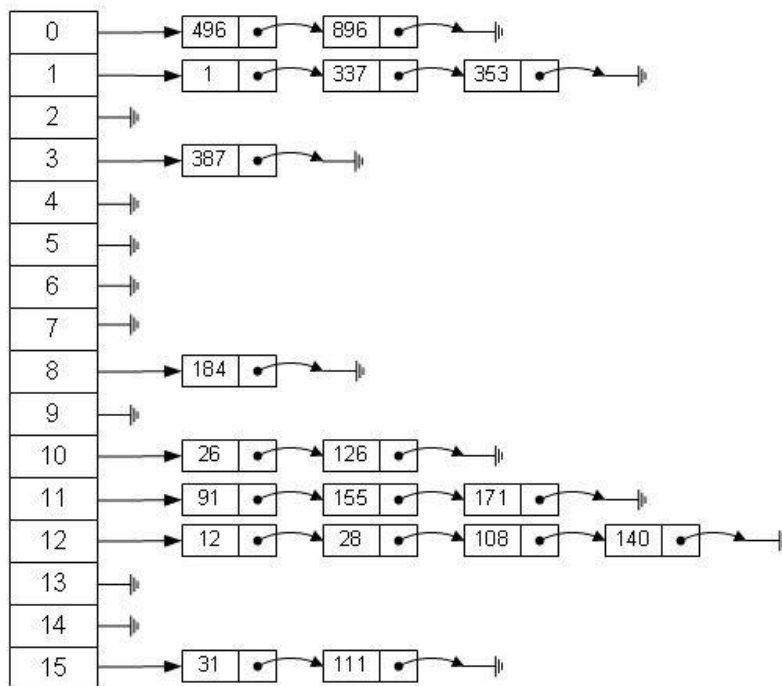
Use the following probe function:

$$p(K, i) = i;$$



$$\text{Pos}(i) = (h(k) + p(k,i)) \% M$$

链地址法：



4、负载因子：  $\alpha = \text{填入表中的元素个数} / \text{散列表的长度}$ 。

5、哈希表扩容：当负载因子过大时触发哈希表扩容操作。

6、常用字符串哈希函数：BKDRHash, APHash, DJBHash, JSHash, RSHash, SDBMHash, PJWHash, ELFHash

例：// BKDR Hash Function

```
unsigned int BKDRHash(char *str)
```

```
{
```

```
    unsigned int seed = 131; // 31 131 1313 13131 131313 etc..
```

```
    unsigned int hash = 0;
```

```
    while (*str)
```

```
    {
```

```
        hash = hash * seed + (*str++);
```

```
    }
```

```
    return (hash & 0x7FFFFFFF);
```

```
}
```