散列函数

1. 散列表

- 散列表既是一种重要的存储方式，也是一种常用的查找方法。
- 理想的情况是，希望不进行关键字间的比较，一下子就能找到所要找的记录。那就必须在记录的关键字和其存储地址之间建立起一个确定的对应关系，使每个关键字都和唯一的储存地址相对应。

2. 散列函数

- 在记录的关键字和其存储地址之间建立的对应关系称为散列函数。散列函数是从关键字空间到存储地址空间的一种映象。
- 哈希函数好坏的主要评价指标是：第一，散列函数构造简单；第二，能“均匀”地将关键字映射到地址空间。

2. 散列函数

- (1) 直接定址法
- 取关键字或关键字的某个线性函数作为散列地址，即：
- H(key)=key，或H(key)=a·key+b (a,b是常数)

2. 散列函数

- (2) 数字分析法
- 对关键字的各位进行分析，去掉分布不均匀的位，留下分布均匀的位作为散列地址。
- 这种方法适用于关键字位数比散列地址位数多，且事先知道关键字所有具体取值的情况。

2. 散列函数

- (3) 平方取中法
- 将关键字平方后，取中间几位作为散列地址。
- 一个数的平方的中间几位和这个数的每一位都有关联，这样由随机分布的关键字得到的散列地址也是随机分布的。这种方法适用于事先不知道关键字所有具体取值的情况。

2. 散列函数

- (4) 折叠法
- 将关键字分割成位数相等的几部分(最后一部分可以例外)，然后取这几部分的和作为散列地址。
- 这种方法适用于关键字位数较多，且每一位上数字分布都比较均匀的情况。

2. 散列函数

- (5) 除余法
- 取关键字除以某个不大于散列表长度m的数p所得的余数作为散列地址，即：
- H(key)=key MOD p (p≤m)
- 这里p一般选素数。

2. 散列函数

- (6) 随机数法
- 取关键字的随机函数值作为散列地址，即：
- H(key)=random(key)
- 这种方法比较适用于关键字长短不一的情况。

思考

- 除余法中的除数p为什么一般选素数？