

数据结构和算法

作者: 小甲鱼

让编程改变世界

Change the world by program

构造散列函数的两个基本原则

计算
简单



分布
均匀



好的散
列函数



直接定址法

- 例一：有一个从1到100岁的人口数字统计表，其中，年龄作为关键字，哈希函数取关键字自身。
- 即： $f(\text{key}) = \text{key}$

地址	年龄	人口
00	0	500万
01	1	650万
02	2	450万
.....
20	20	1500万
.....

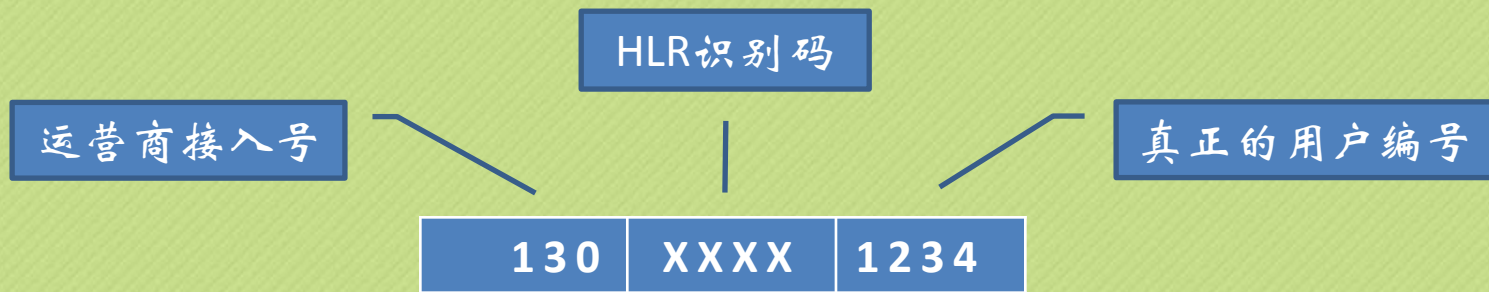
直接定址法

- 例二：如果现在要统计的是1980年以后出生的人口数，那么我们对出生年份这个关键字可以变换为：用年份减去1980的值来作为地址。
- 即： $f(\text{key}) = \text{key} - 1980$

地址	出生年份	人口
00	1980	1500万
01	1981	1600万
02	1982	1300万
.....
20	2000	1500万

数字分析法

- 数字分析法通常适合处理关键字位数比较大的情况，例如我们现在要存储某家公司员工登记表，如果用手机号作为关键字，那么我们发现抽取后面的四位数字作为散列地址是不错的选择。



平方取中法

- 平方取中法是将关键字平方之后取中间若干位数字作为散列地址。



折叠法

- 折叠法是将关键字从左到右分割成位数相等的几部分，然后将这几部分叠加求和，并按散列表表长取后几位作为散列地址。



除留余数法

- 此方法为最常用的构造散列函数方法，对于散列表长为m的散列函数计算公式为：
— $f(\text{key}) = \text{key} \bmod p (p \leq m)$
- 事实上，这个方法不仅可以对关键字直接取模，也可以通过折叠、平方取中后再取模。
- 例如下表，我们对有12个记录的关键字构造散列表时，就可以用 $f(\text{key}) = \text{key} \bmod 12$ 的方法。

下标	0	1	2	3	4	5	6	7	8	9	10	11
关键字	12	25	38	15	16	29	78	67	56	21	22	47

除留余数法

- p的选择是关键，如果对于这个表格的关键字，p还选择12的话，那得到的情况未免也太糟糕了：

下标	0	0	0	0	0	0	0	0	0	0	0	0
关键字	12	24	36	48	60	72	84	96	108	120	132	144

- p的选择很重要，如果我们把p改为11，那结果就另当别论啦：

下标	1	2	3	4	5	6	7	8	9	10	0	1
关键字	12	24	36	48	60	72	84	96	108	120	132	144

随机数法

- 选择一个随机数，取关键字的随机函数值为它的散列地址。
- 即： $f(\text{key}) = \text{random}(\text{key})$ 。
- 这里的random是随机函数，当关键字的长度不等时，采用这个方法构造散列函数是比较合适的。

视不同的情况采用不同的散列函数

- 现实中，我们应该视不同的情况采用不同的散列函数，这里给大家一些参考方向：
 - 计算散列地址所需的时间
 - 关键字的长度
 - 散列表的大小
 - 关键字的分布情况
 - 记录查找的频率