# 常见的哈希函数

# 好的哈希函数?

# 哈希函数

一般来说，一个好的哈希函数应满足下列两个条件：

（1）计算简单

（2）冲突少

# 哈希函数

常见的哈希函数构造方法有：

- 直接哈希函数
- 数字分析法
- 平方取中法
- 折叠法
- 除留余数法
- 随机数法

# 解放后每年出生人数的统计：

| 哈希地址 | | | | | | |
|---|---|---|---|---|---|---|
| 出生年份 | 1949 | 1950 | 1951 | …… | 1970 | …… |
| 出生人数 | ×××× | ×××× | ×××× | …… | ×××× | …… |

H（key）=key +（-1948）

# 直接哈希函数：

- 取关键字本身或关键字的某个线性函数值作为哈希地址，
- 即：H（key）=key
- 或　H（key）=a* key+b（a，b为常数）。

## 解放后每年出生人数的统计：

| 哈希地址 | | | | | | |
|---|---|---|---|---|---|---|
| 出生年份 | 1949 | 1950 | 1951 | …… | 1970 | …… |
| 出生人数 | ×××× | ×××× | ×××× | …… | ×××× | …… |

H（key）=key +（-1948）

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 8 | 1 | 3 | 4 | 6 | 5 | 3 | 2 |
| 8 | 1 | 3 | 7 | 2 | 2 | 4 | 2 |
| 8 | 1 | 3 | 8 | 7 | 4 | 2 | 2 |
| 8 | 1 | 3 | 0 | 1 | 3 | 6 | 7 |
| 8 | 1 | 3 | 2 | 2 | 8 | 1 | 7 |
| 8 | 1 | 3 | 3 | 8 | 9 | 6 | 7 |
| 8 | 1 | 3 | 5 | 4 | 1 | 5 | 7 |
| 8 | 1 | 3 | 6 | 8 | 5 | 3 | 7 |
| 8 | 1 | 4 | 1 | 9 | 3 | 5 | 5 |

n=80,d=8,r=10,s=2

1，2，3，8位分布不均匀，不能取。可取第4、6两位组成的2位十进制数作为每个数据的哈希地址，则图中列出的关键字的哈希地址分别为：

**45，72，84，03，28，39，51，65，13**

# 2. 数字分析法

•设n个d位数的关键字，由r个不同的符号组成，此r个符号在关键字各位出现的频率不一定相同，可能在某些位上均匀分布，即每个符号出现的次数都接近于n / r次，而在另一些位上分布不均匀。则**选择其中分布均匀的s位作为哈希地址**，即H（key）=“key中数字均匀分布的s位”

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| ⋮ | | | | | | | |
| 8 | 1 | 3 | 4 | 6 | 5 | 3 | 2 |
| 8 | 1 | 3 | 7 | 2 | 2 | 4 | 2 |
| 8 | 1 | 3 | 8 | 7 | 4 | 2 | 2 |
| 8 | 1 | 3 | 0 | 1 | 3 | 6 | 7 |
| 8 | 1 | 3 | 2 | 2 | 8 | 1 | 7 |
| 8 | 1 | 3 | 3 | 8 | 9 | 6 | 7 |
| 8 | 1 | 3 | 5 | 4 | 1 | 5 | 7 |
| 8 | 1 | 3 | 6 | 8 | 5 | 3 | 7 |
| 8 | 1 | 4 | 1 | 9 | 3 | 5 | 5 |
| ⋮ | | | | | | | |

n=80,d=8,r=10,s=2

1，2，3，8位分布不均匀，不能取。可取第4、6两位组成的2位十进制数作为每个数据的哈希地址，则图中列出的关键字的哈希地址分别为：

**45，72，84，03，28，39，51，65，13**

题目：请为BASIC源程序中的标识符建立一个哈希表。假设BASIC语言中允许的标识符为一个字母，或一个字母加一个汉字。取标识符在计算机中的八进制数为它的关键字。

| 数据 | 关键字 |
|---|---|
| A | 0100 |
| I | 1100 |
| J | 1200 |
| I0 | 1160 |
| P1 | 2061 |
| P2 | 2062 |
| Q1 | 2161 |
| Q2 | 2162 |
| Q3 | 2163 |

解：标识符数量为

26+26*10=286

需要的存储空间为3位8进制或者9位二进制。表中的关键字没有均匀分布，采用平方后的中间3位均匀发布，可以作为哈希地址

# 3. 平方取中法

•取关键字平方后的中间几位作为哈希地址，即哈希函数为：

H（key）= “key$^2$的中间几位”，

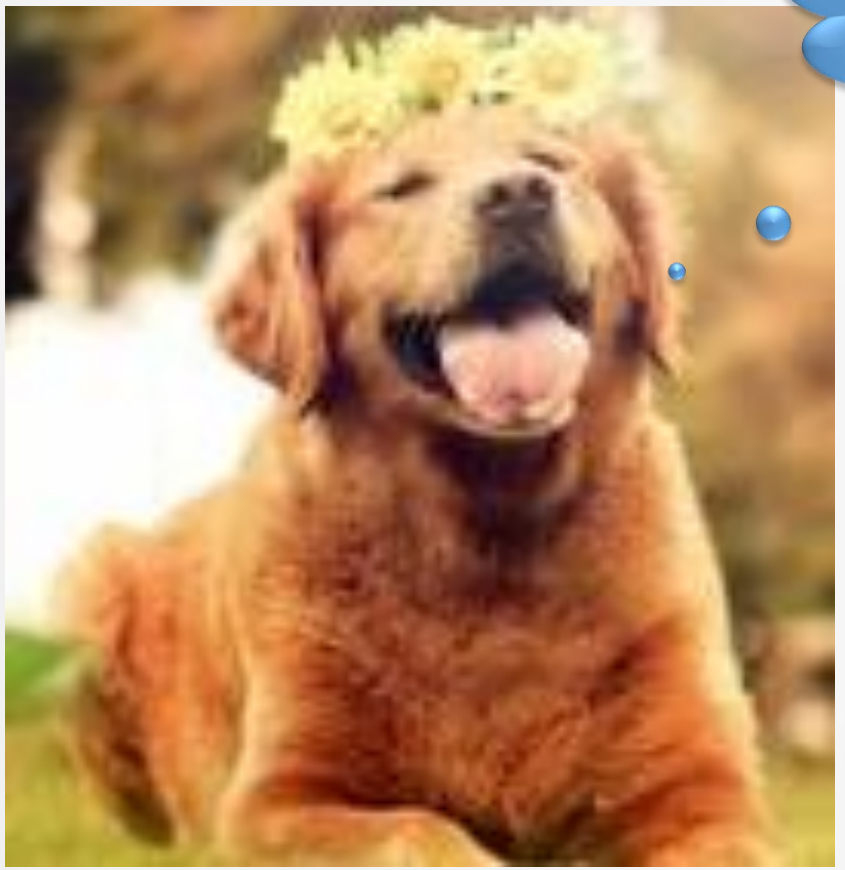
其中，所取的位数由哈希表的大小确定

| 数据 | 关键字 | (关键字)$^2$ | 哈希地址($2^{17}$~$2^9$) |
|---|---|---|---|
| A | 0100 | 0010000 | 010 |
| I | 1100 | 1210000 | 210 |
| J | 1200 | 1440000 | 440 |
| I0 | 1160 | 1370400 | 370 |
| P1 | 2061 | 4310541 | 310 |
| P2 | 2062 | 4314704 | 314 |
| Q1 | 2161 | 4734741 | 734 |
| Q2 | 2162 | 4741304 | 741 |
| Q3 | 2163 | 4745651 | 745 |

# 平方取中法思想

以关键字的平方值的中间几位作为存储地址。求“关键字的平方值” 的目的是“扩大差别”和“贡献均衡”。

即：关键字的各位都在平方值的中间几位有所贡献，Hash值中应该有各位影子。



关键字位数特别多，
怎么办？

# 4. 折叠法

•关键字位数较长时，可将关键字分割成位数相等的几部分（最后一部分位数可以不同），取这几部分的叠加和（舍去高位的进位）作为哈希地址。位数由存储地址的位数确定。叠加时有两种方法：

- 移位叠加法，即将每部分的最后一位对齐，然后相加；
- 边界叠加法，即把关键字看作一纸条，从一端向另一端沿边界逐次折叠，然后对齐相加。

$$\begin{array}{r} d_r\cdots\cdots\ d_2\ \ d_1 \\ d_{2r}\cdots\cdots d_{r+2}\ \ d_{r+1} \\ +)\ d_{3r}\cdots\cdots d_{2r+2}\ d_{2r+1} \\ \hline S_r\cdots\cdots\ S_2\ \ S_1 \end{array}$$

(a) 移位叠加法

$$\begin{array}{r} d_r\cdots\cdots\ d_2\ \ d_1 \\ d_{r+1}\cdots\cdots d_{2r-1}d_{2r} \\ +)\ d_{3r}\cdots\cdots d_{2r+2}\ d_{2r+1} \\ \hline S_r\cdots\cdots\ S_2\ \ S_1 \end{array}$$

(b) 边界叠加法

此方法适合于：关键字的数字位数特别多。

# 5.除留余数法

取关键字被某个不大于哈希表长度m的数p除后的余数作为哈希地址，即：

H（key）=key MOD p(p≤m)

例 p=21

| 关键字 | 28 | 35 | 63 | 77 | 105 |
|---|---|---|---|---|---|
| 哈希地址 | 7 | 14 | 0 | 14 | 0 |

其中p的选择很重要，如果选得不好会产生很多冲突。

比如关键字都是10的倍数，而p=10

# 6. 随机数法

- 选择一个随机函数，取关键字的随机函数值作为哈希地址，
- 即：H（key）=random（key）
- 其中random为随机函数。

实际工作中需根据不同的情况采用不同的哈希函数。通常需要考虑的因素有：

计算哈希函数所需时间；
关键字的长度；
哈希表的大小；
关键字的分布情况；
记录的查找频率。

我有一个电话号码本，怎么根据姓名建立哈希表呢?