

11.2.1 数字关键词的散列函数构造

一个好的散列函数一般应该考虑下列两个因素

1. 计算简单：提高转换速度
2. 关键词对应的地址空间分布均匀：尽量减少冲突

数字关键词的散列函数构造

1. 直接定址法
2. 除留余数法：散列函数为 $h(key) = key \bmod p$
 1. 一般 p 取为散列函数的大小
 2. p 一般取素数（映射均匀）
3. 数字分析法
4. 折叠法（每一位都会对最终的结果产生影响）
5. 平方取中法（每一位都会对最终的结果产生影响）

11.2.2 字符串关键词的散列函数构造

1. 一个简单的散列函数-ASCII码加和法
 1. $h(key) = (\sum key[i]) \bmod TableSize$
 2. 冲突严重！
2. 简单的改进：前三个字符移位法
 1. $h(key) = (key[0] \times 27^2 + key[1] \times 27 + key[2]) \bmod TableSize$
3. 好的散列函数-移位法
 1. 涉及关键词所有的 n 个字符，且分布地很好
 2. 如何快速计算？

```
Index Hash(const char *Key, int TableSize)
{
    unsigned int h=0;
    while(*Key!='\0')
    {
        h=(h<<5) + *Key++;
    }
    return h%TableSize;
}
```