查找

1. 主关键字：能够唯一确定数据元素的关键字。次关键字：不能唯一确定数据元素的关键字

顺序查找

1. 时间复杂度为$O(n)$, $ASL_s = \frac{n+1}{2}$,针对的是等概率的情况
2. 优点：

   算法简单，且适应面广，对表的结构物特殊要求，无论是否按关键字有序排列，均可应用
3. 缺点：平均查找长度大， 查找效率低

索引查找

1. 索引表的构造：$R_1, R_2, \cdots R_L$,要保证R_k块中的所有关键字 $\leq (\geq) R_{k+1}$中的，称为分块有序
2. 然后对每一块建立索引项，索引项要包含：
   1. 关键字项，该块中最大的关键字的值
   2. 指针项，该块第一个记录在表中的位置
3. 查找性能分析，$ASL_{bs} = \frac{1}{2}(\frac{n}{s} + s) + 1$,因此在分块的过程中，要尽量保证$s = \sqrt{n}$，这样才能使查找性能最高

哈希查找

1. 哈希法，又叫杂凑法，散列法，
2. 哈希表，哈希函数
3. 哈希查找的关键
   1. 如何选择哈希函数，尽可能的减少冲突的出现
   2. 当冲突出现时，如何解决冲突
4. 哈希函数应满足以下条件
   1. 简单容易计算
   2. 产生的冲突少
5. 常见的哈希函数
   1. 直接哈希函数
   2. 数字分析法
   3. 平方取中法
   4. 折叠法
      1. 移位叠加法
      2. 边界叠加法
   5. 除留取余法
   6. 随机数法
6. 构造哈希函数的过程中，要考虑以下几点
   1. 是否简单容易计算
   2. 关键字的长度
   3. 哈希表的大小

4. 关键字的分布情况
5. 记录的查找频率

7. 冲突处理的方法
    1. 开放地址法
        1. 线性探测再散列
        2. 二次探测再散列
        3. 伪随机探测再散列
    2. 再哈希法，选用一系列的哈希函数
    3. 链地址法
    4. 公共溢出区法

8. 哈希查找

    查找的过程和哈希表构造的过程几乎一致，给定K值，计算哈希地址，若系位置无数据，则哈希查找失败，若和给定的值不一样，则按照构造哈希表时的冲突处理方法探测下一地址，直到找到某个位置为空或者表中所填的数据和给定值相等结束。

9. 性能分析
    1. 和关键字比较的次数和哈希函数，冲突处理方法，以及哈希表的装填因子有关

        $装填因子\alpha = \frac{表中填入的记录数}{哈希表的长度}$，
    2. 一般情况下，冲突处理方法相同的哈希表，其平均查找长度依赖于哈希表的装填因子α，常见的冲突处理方法的查找成功的平均查找长度为
        1. 线性探测再散列

            $$S_{ns} = \frac{1}{2}(1 + \frac{1}{1-\alpha})$$
        2. 二次探测再散列

            $$S_{nr} = \frac{1}{\alpha}ln(1-\alpha)$$
        3. 链地址法

            $$S_{nc} = 1 + \frac{\alpha}{2}$$

            还有失败的*ASL*
    3. **哈希表的平均查找长度是α的函数，不是n的函数，与n无关**