# 顺序查找：

遍历判断，最直接最好理解的一种

# 二分查找（折半查找）：

数据先排序，不断折半，二次方程求解很像，不多说了

mid=(low+high)/2, 即mid=low+1/2\*(high-low);

# 插值查找：

基于二分查找，二分查找的改进版

主要是求mid值的方法：

mid=low+(key-a[low])/(a[high]-a[low])\*(high-low)，

# 斐波那契查找

思想也是来源于二分查找，不过好像是针对于斐波那契数列的

# 树表查找

<https://www.cnblogs.com/maybe2030/p/4715035.html>

由于树分很多种，其实研究起来也是很费力的，比如有红黑树，平衡二叉树，B树和B+树

先创建树，然后在进行树遍历查找，这也是数据库中常用的实现方式，其实本质的东西也没多少

# 分块查找

分而治之思想：

**算法思想：**将n个数据元素"按块有序"划分为m块（m ≤ n）。每一块中的结点不必有序，但块与块之间必须"按块有序"；即第1块中任一元素的关键字都必须小于第2块中任一元素的关键字；而第2块中任一元素又都必须小于第3块中的任一元素，……

# 哈希查找

这个不说了，常用的map就是这种，除了普通的hash算法，hash算法的痛点在于，当bucket增加时会发生重hash。所以也有人在研究更好的hash算法。

# 倒排索引

倒排索引源于实际应用中需要根据属性的值来查找记录。这种索引表中的每一项都包括一个属性值和具有该属性值的各记录的地址。由于不是由记录来确定属性值，而是由属性值来确定记录的位置，因而称为倒排索引(inverted index)。带有倒排索引的文件我们称为倒排[索引文件](https://baike.baidu.com/item/%E7%B4%A2%E5%BC%95%E6%96%87%E4%BB%B6)，简称[倒排文件](https://baike.baidu.com/item/%E5%80%92%E6%8E%92%E6%96%87%E4%BB%B6/4137688)(inverted file)。

理解为原先有主键和索引查找的方式，转变为有内容查找，适合全文检索，由于内容或者属性内容较多，所以有分词的说法