5/2

平均性能是斐波纳切黄金分割查找更好

B：有序表长度不需要一定要是一个斐波纳切数才行，是可以补齐成为一个斐波纳切数的 补最大的数目直到长度是斐波纳切数

C：最坏情况下斐波纳切查找性能比折半是要差的

已知一个线性表（38，25，74，63，52，48），假定采用散列函数h（key) = key%7 计算散列地址，并散列存储在散列表A[0....6]中，若采用线性探测方法解决冲突，则在该散列表上进行等概率成功查找的平均查找长度为

38%7=3 （第1次出现3，无冲突，放在位置3，查找次数为1）

25%7=4（第1次出现4，无冲突，放在位置4，查找次数为1）

74%7=4（第2次出现4，有冲突，放在位置5，查找次数为2）

63%7=0（第1次出现0，无冲突，放在位置0，查找次数为1）

52%7=3（第2次出现3，有冲突，发现冲突3,4,5，故只能放到6，查找次数为4）

48%7=6 （第1次出现6，有冲突，发现冲突6,1，故只能放到1，查找次数为3）

结果：（1+1+2+1+4+3）÷6=2

在有序表( 7, 13, 33, 87, 99, 97, 117, 123,129,131,137)中，使用二分查找算法查找13时需要的关键字比较次数是 4

第一次：low=0，high=10，mid=5，97>13，high=mid-1

第二次：low=0，high=4，mid=2，33>13，high=mid-1

第三次：low=0，high=1，mid=0，7<13，low=mid+1

第四次：low=1，high=1，mid=1，查找成功

5/3

下列哪个查找最快 哈希表法（散列表）

A：哈希表查找时间复杂度O(1)

B：顺序查找时间复杂度O(n)

C：二分查找时间复杂度O(logn)

D：二叉排序树查找时间复杂度O(logn)

顺序查找的平均时间是多少n/2

其判定树的高度，也就是为最坏一次查找时，需要比较的次数，所以为[log2n+1]

给定一个整数sum,从有N个有序元素的数组中寻找元素a,b,使得a+b的结果最接近sum,最快的平均时间复杂度是O(n)

2个指针从数组的开始位置和结束位置移动

红黑树和avl树都属于自平衡二叉树；

两者查找、插入、删除的时间复杂度相同；

包含n个内部结点的红黑树的高度是o(logn);

TreeMap是一个红黑树的实现，能保证插入的值保证排序

一个文件包含了200个记录，若采用分块查找法，每块长度为4，则平均查找长度为多少

要是有序，可以二分查找，然后确定在那一个块中((n+1)log2(n+1))/n - 1 ，然后在这个块中查找 要是无序，则顺序查找，平均查找长度就是 （1+50）/2=25.5，然后块中查找(1+4)/2=2.5 总共28

对于静态表的顺序查找法，若在表头设置监视哨，则正确的查找方式为

从最后个元素往开始前查找该数据元素 监视哨可以防止越界,可以不必每次查找时都检查是否到末尾了

假设在有序线性表A[1..30]上进行二分查找,则比较五次查找成功的结点数为 15

30-（24-1）=15，n层排满二叉树的节点总数为2n-1

使用KMP算法在文本串S中找模式串P是一种常见的方法。假设S=P={xyxyyxxyx}，亦即将S对自己进行匹配，匹配过程中正确的next数组是

首先求最大相同前缀后缀长度 模式串的各个子串 前缀 后缀 最大公共元素长度 x 空 空 0 xy x y 0 xyx x ， xy x ， yx 1 （ x ） xyxy x ， xy ， xyx y ， xy ， yxy 2 （ xy ） xyxyy x ， xy ， xyx ， xyxy y ， yy ， xyy ， yxyy 0 xyxyyx x ， xy ， xyx ， xyxy ， xyxyy x ， yx ， yyx ， xyyx ， yxyyx 1 （ x ） xyxyyxx x ， xy ， xyx ， xyxy ， xyxyy ， xyxyyx x ， xx ， yxx ， yyxx ， xyyxx ， yxyyxx 1 （ x ） xyxyyxxy x ， xy ， xyx ， xyxy ， xyxyy ， xyxyyx ， xyxyyyxx y ， xy ， xxy ， yxxy ， yyxxy ， xyyxxy ， yxyyxxy 2 （ xy ） xyxyyxxyx x ， xy ， xyx ， xyxy ， xyxyy ， xyxyyx ， xyxyyyxx ， xyxyyyxxy x ， yx ， xyx ， xxyx ， yxxyx ， yyxxyx ， xyyxxyx ， yxyyxxyx 3 （ xyx ） 2、通过“最长相同前缀后缀长度值右移一位，然后初值赋为 -1 ”得到的 next 数组： 模式串 X Y X Y Y X X Y X 前缀最大公共元素 0 0 1 2 0 1 1 2 3 Next -1 0 0 1 2 0 1 1 2