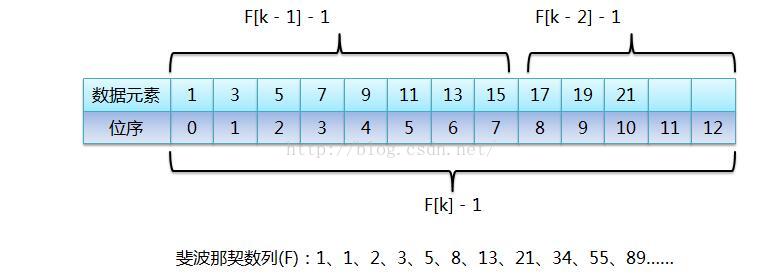
斐波那契数列，又称黄金分割数列，指的是这样一个数列：1、1、2、3、5、8、13、21、····，在数学上，斐波那契被递归方法如下定义：F(1)=1，F(2)=1，F(n)=f(n-1)+F(n-2) （n>=2）。该数列越往后相邻的两个数的比值越趋向于黄金比例值（0.618）。

     斐波那契查找就是在二分查找的基础上根据斐波那契数列进行分割的。在斐波那契数列找一个等于略大于查找表中元素个数的数F[n]，将原查找表扩展为长度为F[n](如果要补充元素，则补充重复最后一个元素，直到满足F[n]个元素)，完成后进行斐波那契分割，即F[n]个元素分割为前半部分F[n-1]个元素，后半部分F[n-2]个元素，找出要查找的元素在那一部分并递归，直到找到。

斐波那契查找的时间复杂度还是O(log 2 n )，但是 与折半查找相比，斐波那契查找的优点是它只涉及加法和减法运算，而不用除法，而除法比加减法要占用更多的时间，因此，斐波那契查找的运行时间理论上比折半查找小，但是还是得视具体情况而定。

对于斐波那契数列：1、1、2、3、5、8、13、21、34、55、89……（也可以从0开始），前后两个数字的比值随着数列的增加，越来越接近黄金比值0.618。比如这里的89，把它想象成整个有序表的元素个数，而89是由前面的两个斐波那契数34和55相加之后的和，也就是说把元素个数为89的有序表分成由前55个数据元素组成的前半段和由后34个数据元素组成的后半段，那么前半段元素个数和整个有序表长度的比值就接近黄金比值0.618，假如要查找的元素在前半段，那么继续按照斐波那契数列来看，55 = 34 + 21，所以继续把前半段分成前34个数据元素的前半段和后21个元素的后半段，继续查找，如此反复，直到查找成功或失败，这样就把斐波那契数列应用到查找算法中了。



从图中可以看出，当有序表的元素个数不是斐波那契数列中的某个数字时，需要把有序表的元素个数长度补齐，让它成为斐波那契数列中的一个数值，当然把原有序表截断肯定是不可能的，不然还怎么查找。然后图中标识每次取斐波那契数列中的某个值时(F[k])，都会进行-1操作，这是因为有序表数组位序从0开始的，纯粹是为了迎合位序从0开始。