1.什么是数据结构  
 （1）数据结构是计算机存储、组织数据的方式。指相互之间存在一种或多种特定关系的数据元素的集合  
  
  
  
  
2.什么是算法  
 （1）算法就是定义良好的计算过程，它去一个或者一组的值为输入，并产生一个或者一组的值为输出。简单来说，算法就是一系列的计算步骤，用来将输入数据转化成输出结果  
  
  
  
  
3.剑指offer在初阶学完就可以看

1.算法的复杂度

（1）算法在编写成可执行程序后，运行时需要浪费时间和空间。算法的好坏也是从时间和空间两个方面来衡量的-----也就是时间复杂度和空间复杂度

《1》时间复杂度主要衡量一个算法运行的快慢，空间复杂度主要衡量一个算法所需要的额外空间

《2》不过由于现在技术，现在已经不是很关注算法的空间复杂度了

2.1算法的时间复杂度

（1）时间复杂度的定义：算法的时间复杂度是一个函数，它定量的描述了算法运行所需要的时间

（2）算法中的基本操作的执行次数就是时间复杂度 -----也就是说找到某个语句与问题规模N之间的数学表达式，也就是找到了时间复杂度

（3）同一个算法在不同的环境下所需要的时间也可能是不同的 -----也就是说没有办法确定一个算法具体能在多少时间运行出来

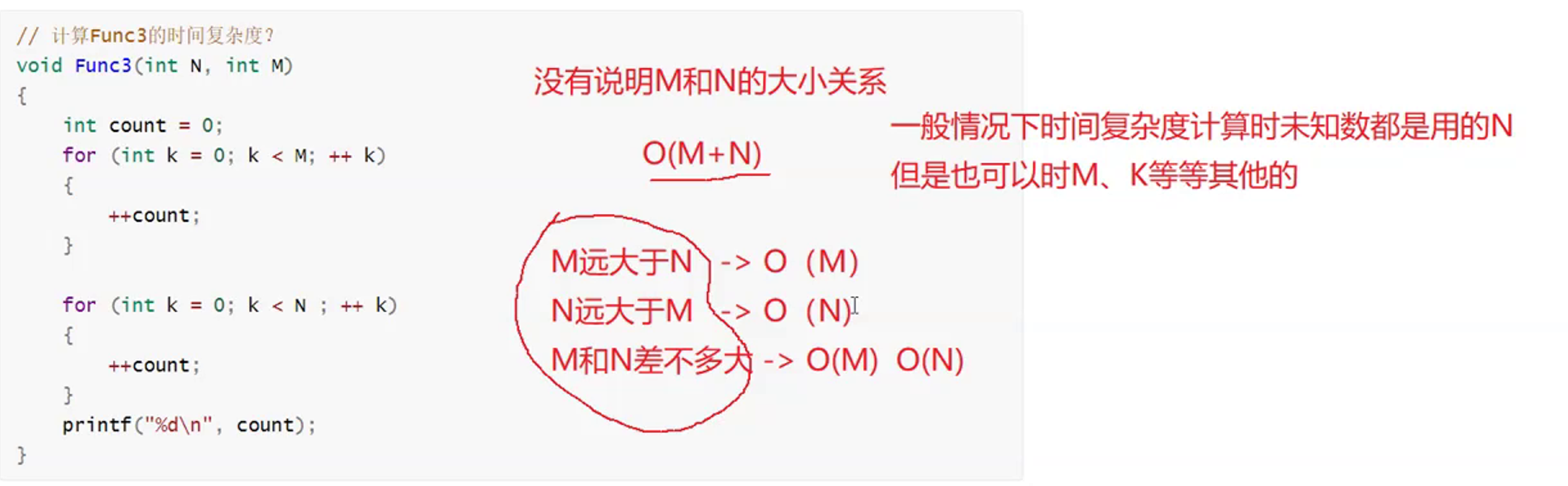


（4）我们在计算语句的执行次数并不用精确的写出，只需要知道大概执行次数，我们以后就要用到大O表示的渐进表示法

这就是表示的形式，括号里面的是变化速度最快的那一项

2.2大O的渐进表示法的规则

（1）部分图例式样



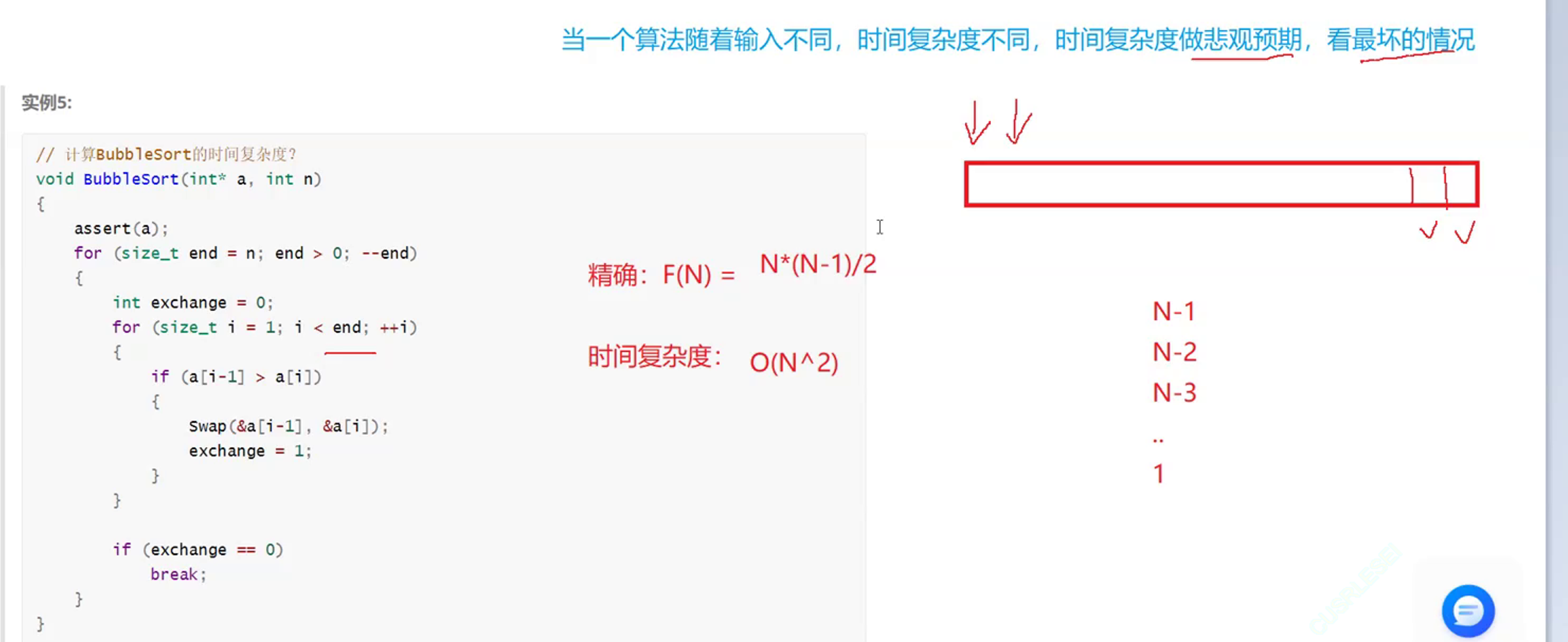
（2）



《1》常见的平均时间复杂度基本都是（最好+最坏）/2

《2》时间复杂度里面的N前面的系数都要去掉

（3）



（4）比较重要的折半查找为什么的时间复杂度为什么是log2 N （这个是最坏的情况）

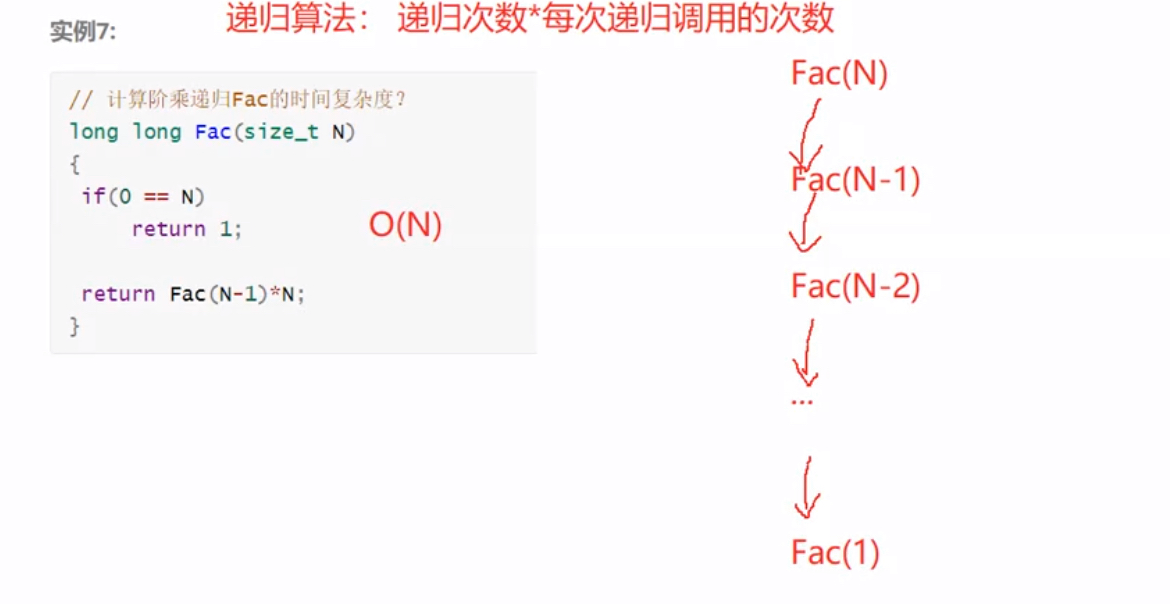
《1》第一种方法是倒推法，在找到那个数字的时候是经历了x次的对半（还剩下最后一个数，也是刚好要找的那个数），也就是除以2。 反过来想就是将所剩的数字个数乘以x个2，就是全部的数字N。也就是 log以2为底N的对数次。

《2》

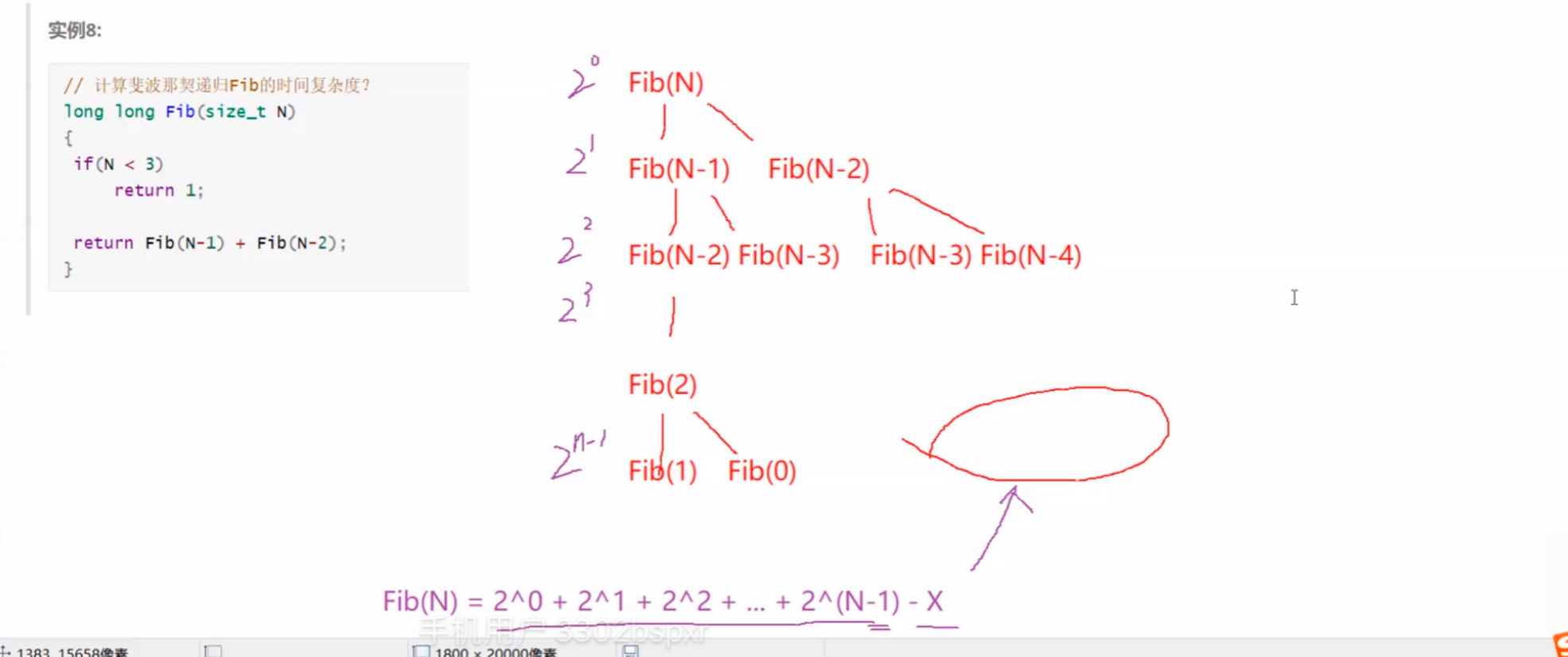


这是第二种方法推导，这个是正向的：N个数要除以x个2才等于1，也就是1\*2\*2\*2...=N

（5）递归算法时间复杂度：递归次数\*每次递归调用的次数



(6)斐波那契数列的时间复杂度



//这个时间复杂度就是O(2^n)

3.空间复杂度

（1）也是一个数学的函数表达式。

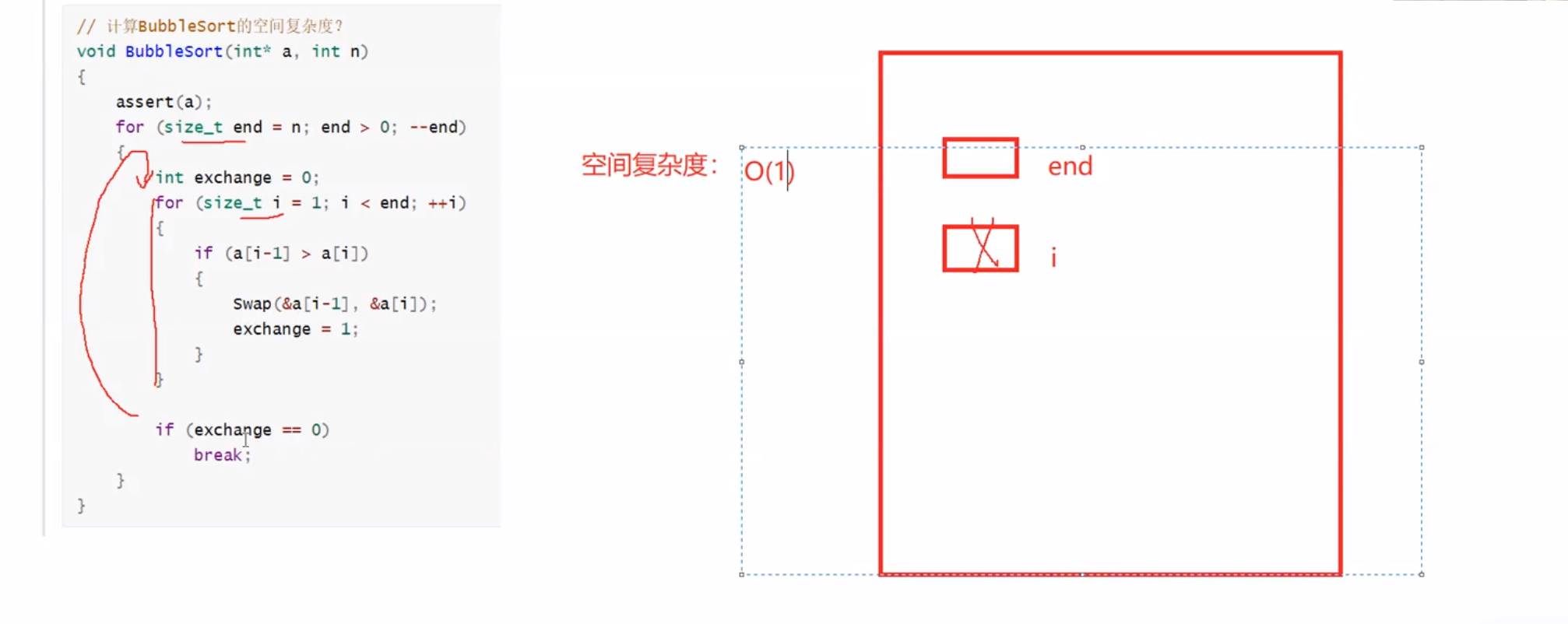
（2）代表一个算法临时占用存储空间大小的量度

（3）空间复杂度算的是变量的个数

（4）采用的也是大O渐进表示法

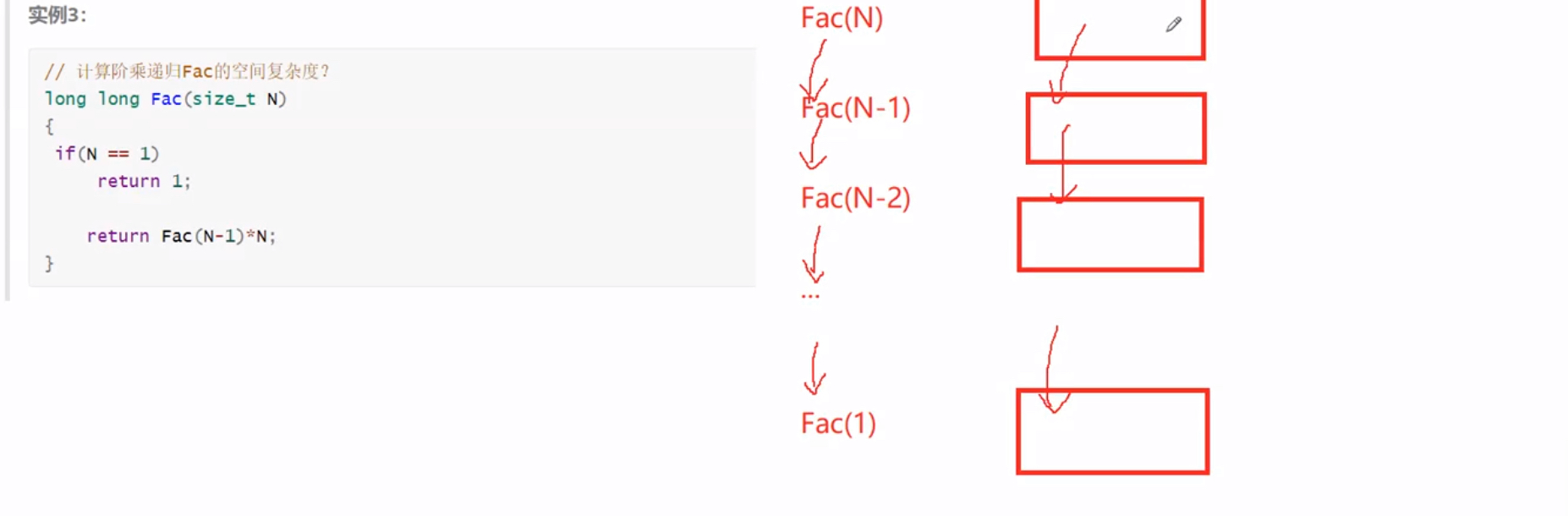
（5）函数运行时需要的栈空间（存储参数，局部变量，一些寄存器信息等）在编译期间已经确定好了，因此空间复杂度主要通过函数在运行时显式申请额外的空间来确定

（6）



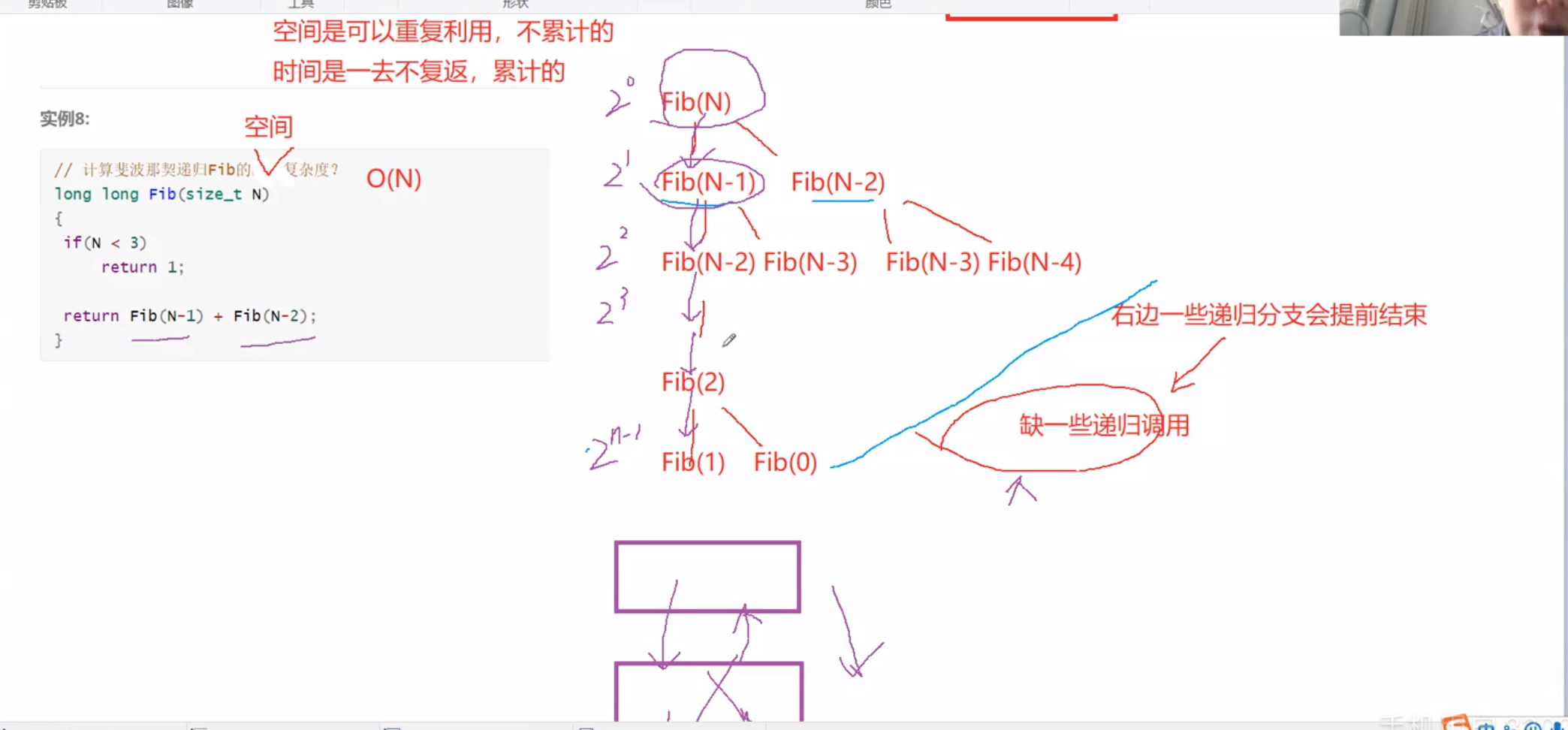
//额外的空间就是临时申请的变量。比如循环的end和i。下面的i每次所用的空间是同样的还有exchange。 -----

（7）



//递归的空间复杂度看的是调用的深度 这个O(N)

(8)

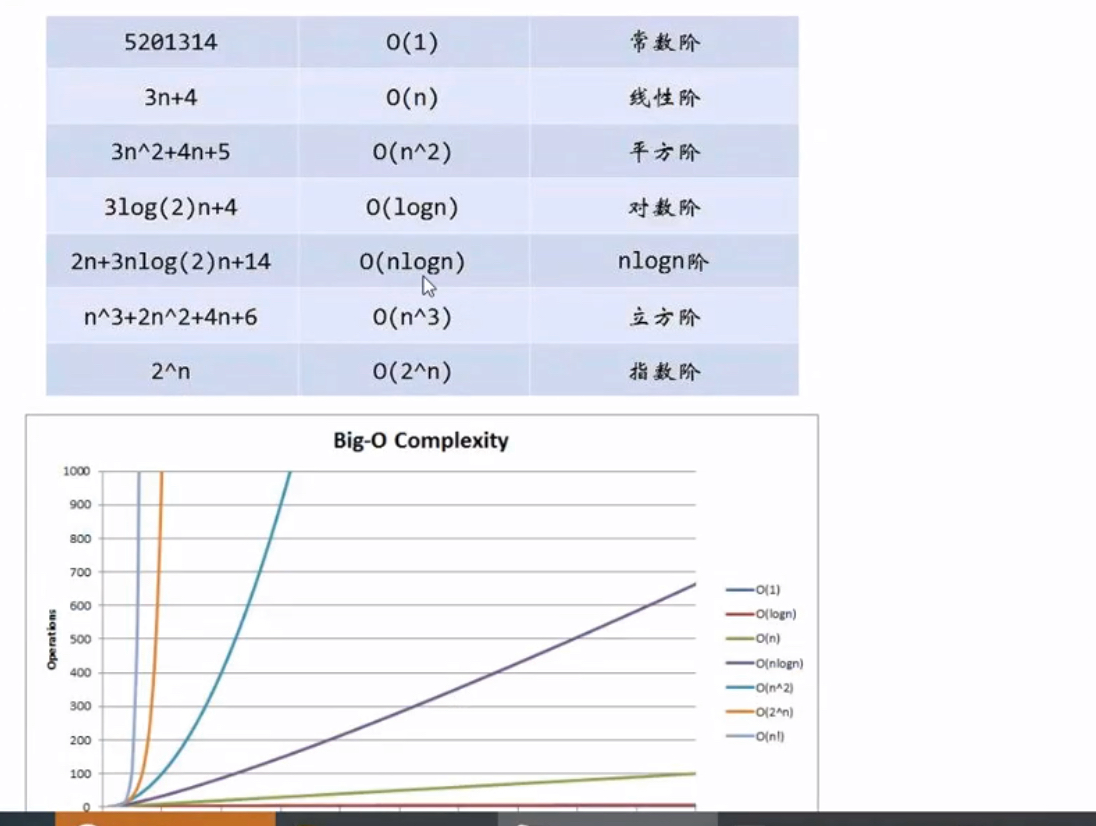


//递归的空间复杂度就是调用最深的那个层次。

//时间不能重复利用，但是空间可以。 最左侧的空间在使用过后会销毁，但是右侧在调用的时候还是用的这些空间

4.复杂度的对比

（1）



5.实践题目

力扣189