

空间复杂度

1.什么是空间复杂度？

表示算法的存储空间与数据规模之间的增长关系。

```
def test(num):  
    total = 0          //占用空间  
    for i in range(num):  
        total += i  
    return total
```

`total`变量会占存储空间，而`for`循环不会占用存储空间，不会申请空间来存储数据，`total`为一个`int`型的数字，不论传入的参数`num`为多少，`total`所占的空间就是一个数字，一个`int`型变量的大小，因此这段代码的空间复杂度就是 $O(1)$ 。

```
def test(nums):        //nums可能为一个list、array  
    array = []          //占用空间 array为一个list  
    for num in nums:  
        array.append(num) //将nums中的每一个值都放入array  
    return array
```

array的空间大小取决于nums的大小，所以array的空间大小和nums的空间大小成正比，因此，这段代码的空间复杂度就是 $O(N)$ 。

以上两个例子可以说明，空间复杂度，在代码中主要看变量。如果一个变量的大小等于常量的话，不论传入的值是多少，那么它的空间复杂度就是 $O(1)$ 。如果一个变量是array, linklist...,它随着输入值的改变而改变的时候，它的空间复杂度就是 $O(N)$ 。

2.常用的空间复杂度

$O(1)$ 、 $O(N)$ 、 $O(N^2)$ （较少） $O(\log N)$, $O(N \log N)$ （一般用不到）

一般算空间复杂度，在代码中寻找变量，如上。另一种是查看递归，递归中的每一层信息都会保存到递归**stack**中，因此递归一般都会用到一个 $O(N)$ 的空间复杂度，哪怕它没有用到任何的变量

3.总结：

$O(1) < O(N) < O(N^2)$

时间复杂度和空间复杂度两者只能二选一，要么以时间换空间，要么以空间换时间

面试时：一般两者都进行考虑

工作时：时间 $>$ 空间