## 面试题

### 1、字节跳动：

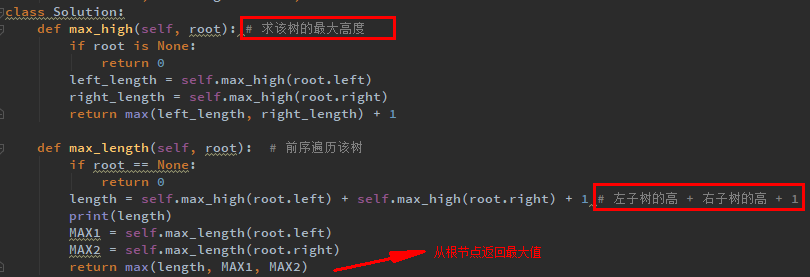
### （1）求树的最大长度之和

找出这棵树的最大长度

Class Node{ Node left; Node right; ...}  
    输入： rootNode  输出： maxLengthPath  
    示例： 最长路径为 61 -> 51 -> 41 -> 4 -> 2 -> 5 -> 42 -> 53 -> 63 ,9个节点，maxLength=9  
                  1  
               /       \  
              2          3  
            /   \        /   \  
           4     5    6     7  
         /       /   
       41     42    
      /  \        \   
     51 52     53  
    /              /   
   61           63

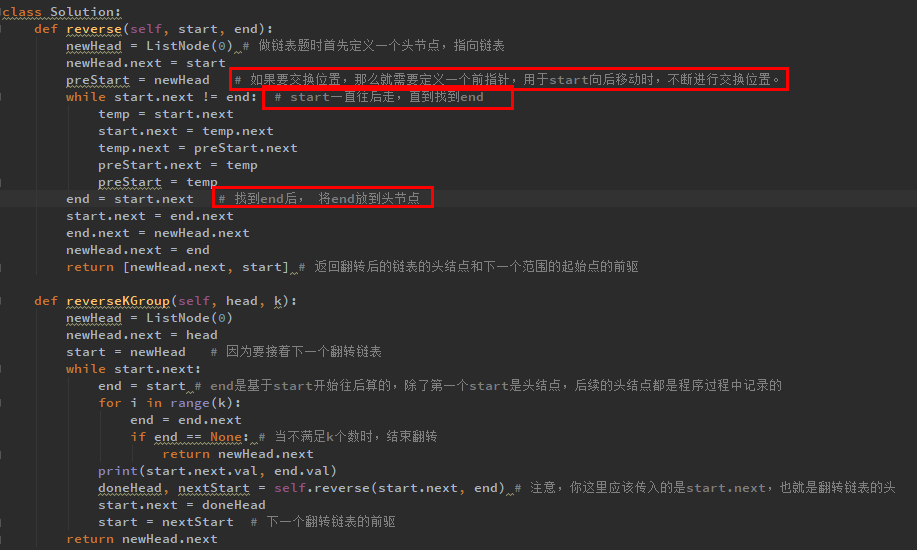
我的思路：遍历每个节点，求出以每个节点为根节点的树的最大长度，然后比较得到最大长度。

关键：最大长度 = 左子树的高 + 右子树的高 + 1



### （2）每n个数交换一次

跟109题有着相似之处，109题是每隔n个数翻转一次



下面是每四个数翻转一次的情况



### （3）在一个二维数组中，每一行都按照从左到右递增的顺序排序，每一列都按照从上到下递增的顺序排序。请完成一个函数，输入这样的一个二维数组和一个整数，判断数组中是否含有该整数。



从最后一行的第一个开始查找, 而不是从第一行的最后一个查找

### （4）我们可以用2\*1的小矩形横着或者竖着去覆盖更大的矩形。请问用n个2\*1的小矩形无重叠地覆盖一个2\*n的大矩形，总共有多少种方法？写出您的代码



### 2、阿里

### （1）拥塞窗口

/\*\*

\* 场景描述

\* 信号灯路口每2分钟计算出一个拥堵指数值，当拥堵指数大于2的时候，认为路口发生异常。

\* 一天之内，上一个异常时段的结束时间到下一个异常时段的开始时间间隔小于10分钟则将两个异常时段合并。

示例结果

\* inter\_id time\_period ds

\* 1 [2016-12-18 00:02:00,2016-12-18 00:18:00) 20161218

\* 1 [2016-12-18 00:28:00,2016-12-18 00:28:00) 20161218

\*

\* 使用任意一种编程语言来实现核心处理逻辑

\*/

class Solution():

def Last(self, List, pos):

Last = pos - 5 if pos >= 5 else 0

for i in range(Last, pos):

temp = List[Last][1]

if List[i][1] > 2:

return True

return False

def period(self, List):

stack = []

for i in range(len(List)):

if List[i][1] > 2:

if self.Last(List, i):

stack[-1][1] = List[i][0] + 2

else:

stack.append([List[i][0], List[i][0] + 2])

return stack

if \_\_name\_\_ == '\_\_main\_\_':

solution = Solution()

List = [

[0, 0.4],

[2, 2.6],

[4, 2.1],

[6, 1.4],

[8, 0.7],

[10, 0.8],

[12, 2.9],

[14, 1.3],

[16, 2.2],

[18, 1.6],

[20, 0.3],

[22, 0.7],

[24, 0.9],

[26, 1.1],

[28, 2.3],

[30, 1.7]

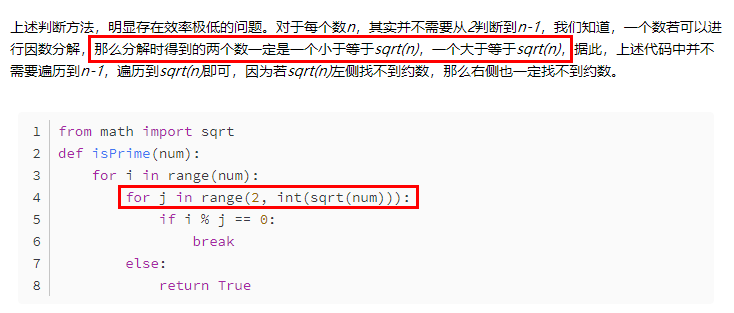
#...

]

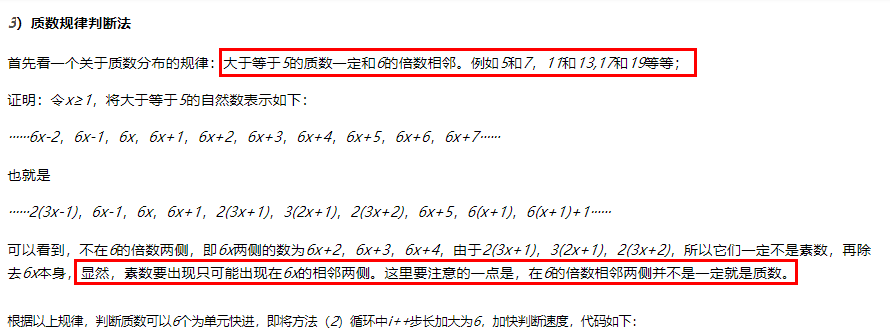
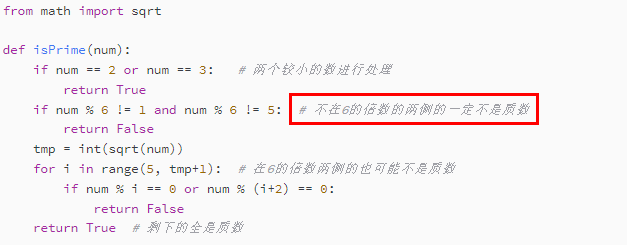
print(solution.period(List))

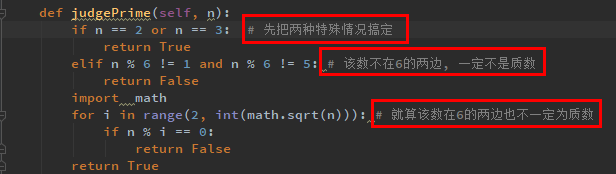
## 经典算法

### 判断质数

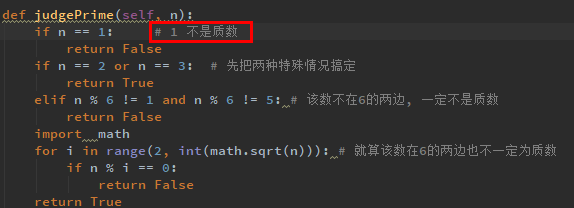


判断一个数是否为质数，就是看这个数是否能够因式分解。

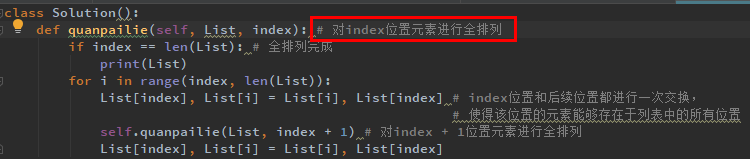


改进如下：

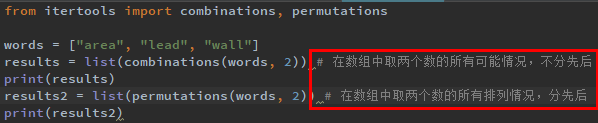


### 全排列

# index位置和后续位置都进行一次交换，使得该位置的元素能够存在于列表中的所有位置



利用permutation函数进行全排列：





### 判断一个数的阶乘的末尾有几个0

我的思路：能产生0的质数组合只有2\*5，然后问题就转变成了对N!进行质数分解后，一共有几个5，因为2的个数显然多于5。比如计算25!的末尾0的个数，包含5的数有5,10,15,20,25,其中25中包含两个5，所以一共包含6个5,25!的末尾有6个0。

