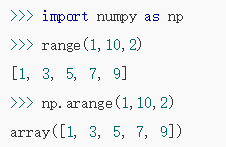
**1. Two sum问题（题号1，easy）**

**（1）range & arange**

range(start, end, step)，返回一个list对象，起始值为start，终止值为end，但不含终止值，步长为step。只能创建int型list。

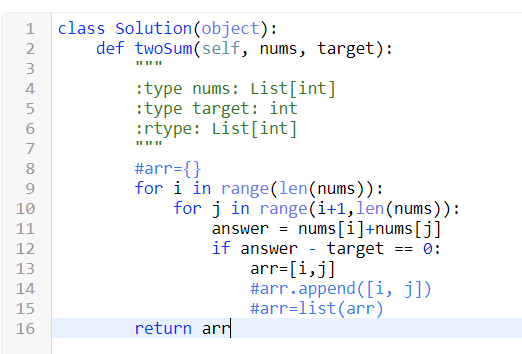
arange(start, end, step)，与range()类似，但是返回一个array对象。需要引入import numpy as np，并且arange可以使用float型数据。



**（2）我的方案**

思路：用每一个数字和其后边所有数字以此相加，找到目标值，返回此时的索引号

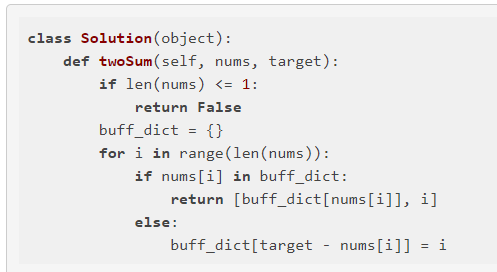
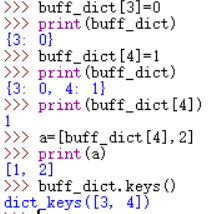
考虑两点：一是不能和自身相加，因此j+1，二是可能有多种结果，所以append;



遗留问题：若用append结果是[[]]形式，怎么把二维数组转一维，但若不考虑重复，可以按上述来写；

运行时间：32ms

**（3）题中方案：**

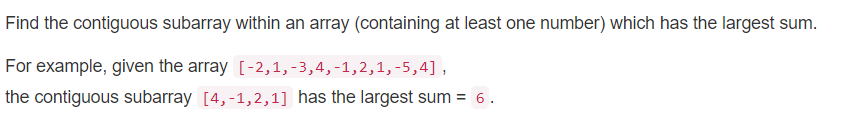
时间：26ms

Nums=[3,2,4], target = 6;

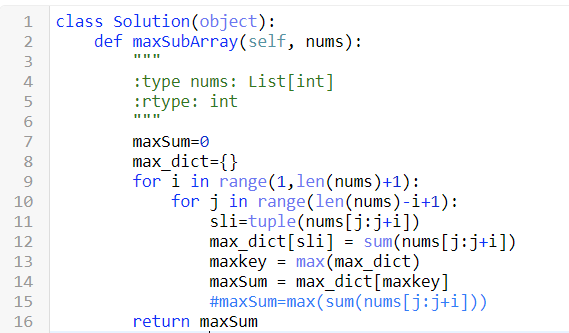
理解：在一开始啥都没有时候创建一个字典，就是说我第一个数和目标差3，那么我就把差的3和我本来3的索引（0）做成一个字典，当后边出现3时，字典就可以查到这个3，然后告诉他前边有个索引是0的3在等它，然后返回这两个3的索引；以此类推，2进来时，告诉他他和目标差4，那么当后边遇到4时，就告诉他前边有个2在等它，把他俩的坐标返回即可，这种方法的好处是不用逐个尝试，前边有记录谁缺什么

**2. Maximum subarray问题（题号53，easy**）

问题：求连续的最大和：

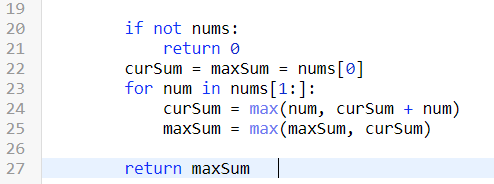


1. 初始思路：创建一个字典，使连续子列表作为键，其和作为值，max这个字典，即可得到其最大值对应的键，但是行不通，因为列表是不可哈希的，因此不可以作为键，大多数的python对象都可以为键，但是列表和字典都是不可哈希的，不能为键，因此可以改为元组



错误：理解问题：输出最大值即可，不需要子列表，因此14行又取了一遍字典里的值，但是结果仍有问题，此时结果为后六个数之和5，i,j有问题

1. 题中方案：



这个思路就是相当于把前两个数先相加，结果与第二个数比，若比它小，则相当于拖后腿，放弃第一个数，从第二个数开始往后加，若比第三个数大，则保留这两个的结果，接着往后加第四个数，结果若比第四个数小则放弃前三个，从第四个数开始加，也就是说从前往后依次相加，每次加两个数，把和与后一个数相比，若大则留作为新的起点，若小则弃，并使后一个数变成新的起点，再将新的数与后一个相加，同理，同时把新的数与和的最大值保存下来，以确定到哪里停止，直至加到最后一个数。

这个方法的优点也在于不用逐个相加，而是先分析数据间的关系，始终排除前边拖后腿的数，并在收集过程中的最值，以便取出，不能不取最值一直加到最后，那样和值有可能变小。