题目：

Given an array S of n integers, find three integers in S such that the sum is closest to a given number, target. Return the sum of the three integers. You may assume that each input would have exactly one solution.

For example, given array S = {-1 2 1 -4}, and target = 1.

The sum that is closest to the target is 2. (-1 + 2 + 1 = 2).

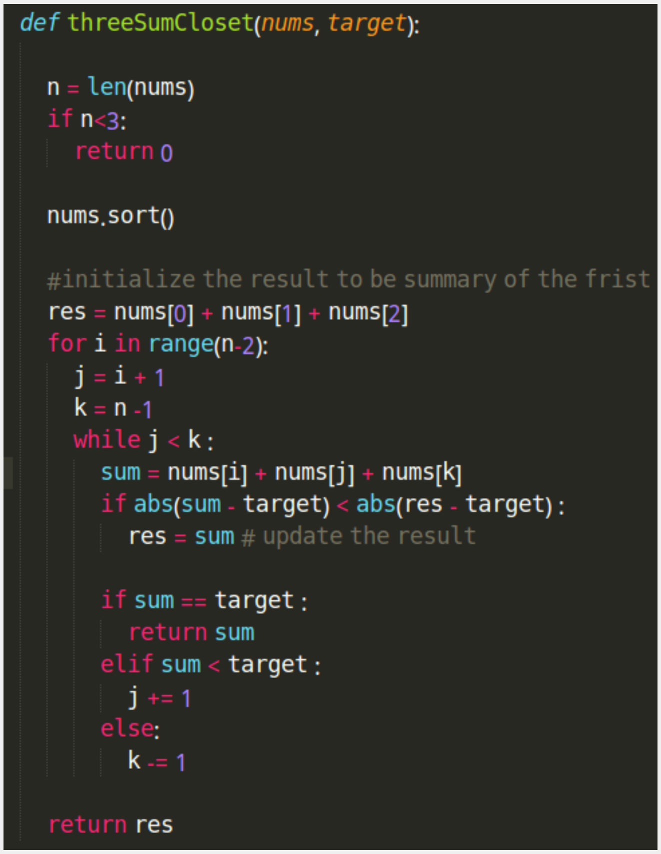
这条题目与3Sum类似，使用三指针法。唯一不同的是需要初始化一个result 。把res初始化为最开头的三个数组元素的和。然后在每次移动指针后把计算结果与这个结果像比较，进而判断是否需要更新这个结果。待三个指针扫过所有的元素后，返回这个结果值即可。

伪代码：

```
res = nums[0] + nums[1] + nums[2]
移动指针 i, j, k
sum = nums[i] + nums[j] + nums[k]
if |sum-target| < |res - target|
        res = sum
if sum == target
        return res
else if sum < target
        移动指针 j ++
else sum > target
        移动指针 k--

return res
```

代码如下



```python
def threeSumCloset(nums, target):

    n = len(nums)
    if n<3:
        return 0

    nums.sort()

    #initialize the result to be summary of the frist
    res = nums[0] + nums[1] + nums[2]
    for i in range(n-2):
        j = i + 1
        k = n -1
        while j < k :
            sum = nums[i] + nums[j] + nums[k]
            if abs(sum - target) < abs(res - target) :
                res = sum # update the result

            if sum == target :
                return sum
            elif sum < target :
                j += 1
            else:
                k -= 1

    return res
```