class Solution:

def maxSubArray(self, nums: List[int]) -> int:

if not nums:

return 0

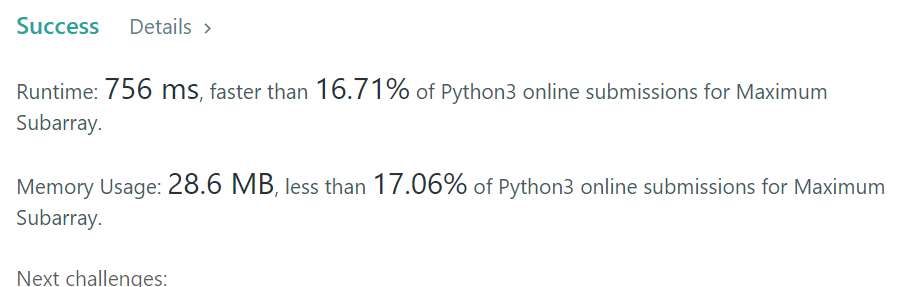
curSum = maxSum = nums[0]

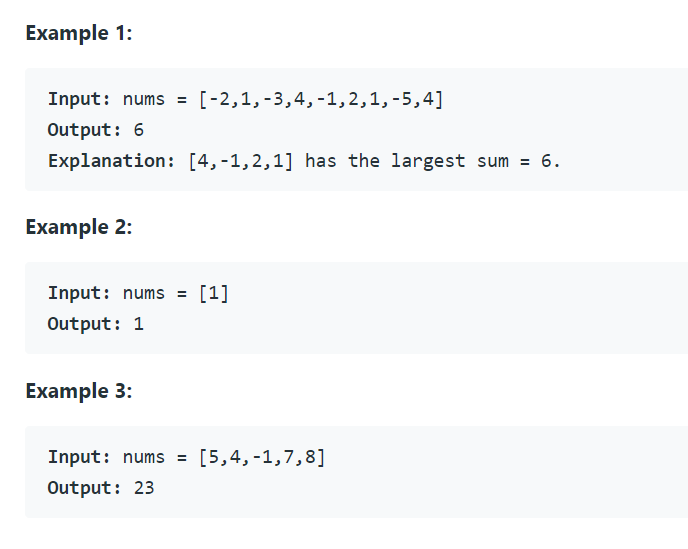
for num in nums[1:]:

curSum = max(num, curSum + num)

maxSum = max(maxSum, curSum)

return maxSum





想法是有兩個紀錄的變數一個是目前某一段sum 保留下一個值與 下一個值+前一段值比較 的max

另一個是目前max sum 與這一段sum的 max比較

kadanes-algorithm跟這個想法依樣

<https://www.codingninjas.com/blog/2020/09/17/a-quick-look-at-kadanes-algorithm/>

噁心的做法

for i in range(1, len(nums)):

if nums[i-1] > 0:

nums[i] += nums[i-1]

return max(nums)

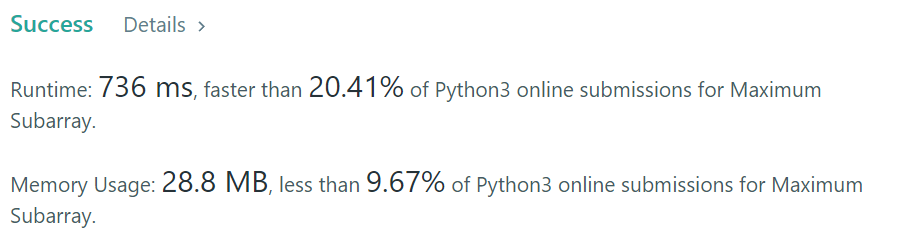
假如前一項是正的 就把它加起來 成為新的自己

下一項就會變成前兩項 加上現在這一項 的和(假如三項都>0)

[-2,1,-3,4,-1,2,1,-5,4]

->[-1,1,-2,4,3,5,6,1,5]

Max = 6



真厲害 我知道一段是正的該往後加 但不知道怎麼跟負數組合或刪除

要多看幾次