*1、请写一个函数*

*def solution(A)*

*对于给定的序列S，返回它的最大子序列之和。*

*一个序列S的子序列是指序列S中拥有连续的数组下标的元素所组成的序列。*

*例如，对于给定的序列*

*A[0]=3 A[1]=2 A[2]=-6 A[3]=4 A[4]=0*

*我们可以得到下面的子序列：*

*[2,-6,4,0] [3,2,-6,4,0] [2] [3,2,-6] [-6,4] [](空的子序列)*

*以及其他的子序列，子序列[]被称为空的子序列，因为其中不包含任何的元素。下面的序列则不是给定序列的子序列S：*

*[3,-6,0],[1],[3,2,-6,0]*

*最大的子序列之和是指一个序列中所有非空的子序列的元素的总和的最大值，用更精确的方式来表示：*

*Max(S[p] + S[p+1] + … + S[q] {其中p,q都是整数，并且} p<q p<q)*

*对于上面的序列S，最大的子序列之和为*

*5 = S[0] + S[1]*

*假定：*

*· N 是 [1..1,000,000]内的整数；*

*· 数组A每个元素是取值范围 [-1,000,000..1,000,000]内的整数；*

*· the result will be an integer within the range [-2,147,483,648..2,147,483,647].*

*复杂度：*

*· 最坏情况下，期望的时间复杂度是O(N);*

*· 最坏情况下，期望的空间复杂度是O(N)，输入存储除外（不计输入参数所需要的存储空间）.*

**答：**#!usr/bin/env python

#encoding:utf-8

'''''

功能：最大子序列问题

'''

def test\_func(num\_list):

'''''

求数组中最大子序列的和，子序列可以不连续

（也可以写成if判断语句只累加整数即可）

'''

length=len(num\_list)

dp=[0]\*length

dp[0]=num\_list[0]

for i in range(length):

dp[i]=max(dp[i-1]+num\_list[i], dp[i-1])

print dp[-1]

def test\_func2(num\_list):

'''''

求数组中最大子序列的和，子序列必须连续

'''

length=len(num\_list)

max\_value=-10000000000

tmp=0

for i in range(length):

tmp=max(tmp+num\_list[i], num\_list[i])

max\_value=max(max\_value, tmp)

print max\_value

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

num\_list=[-4 , 3 ,56 , -15 , 34 , 0 , -14 , 4]

test\_func(num\_list)

print '----------------------------------------------------'

test\_func2(num\_list)

2、一个二进制数K用稀疏数组表示为A，长度为N

写一个函数

def solution(A):

返回k乘以3之后的数组A’的长度。

例如：指定 N = 3

A = [1,4,5]

那么 k = 二进制(110010) =十进制(50)，那么乘以3之后为十进制150，二进制为10010110，

A’ = [1,2,4,7]

那么A’长度为4.

要求：

复杂度越小越好。