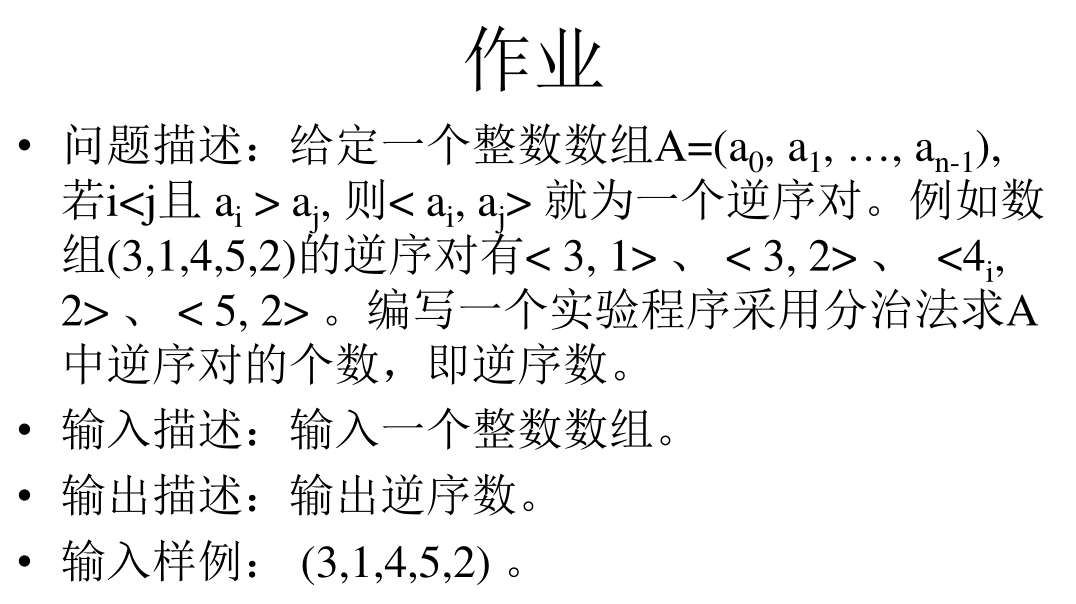
# 作业02

问题描述：



算法思想：采用分治法，将问题转化为子问题，取划分界限点为传入数组A中点，将数组A划分为B和C，先求B中数字与C中数字构成逆序对的个数，再分别求B，C中对应逆序数。其中，当分治法进行到数组A的长度小于等于1时，此时数组内无逆序对数且无法划分，此时返回0。

算法复杂度分析：Ο(nlog2n)。

算法流程图：

开始

输入M,初始化dp数组和min\_id列表

判断len(A)<=1

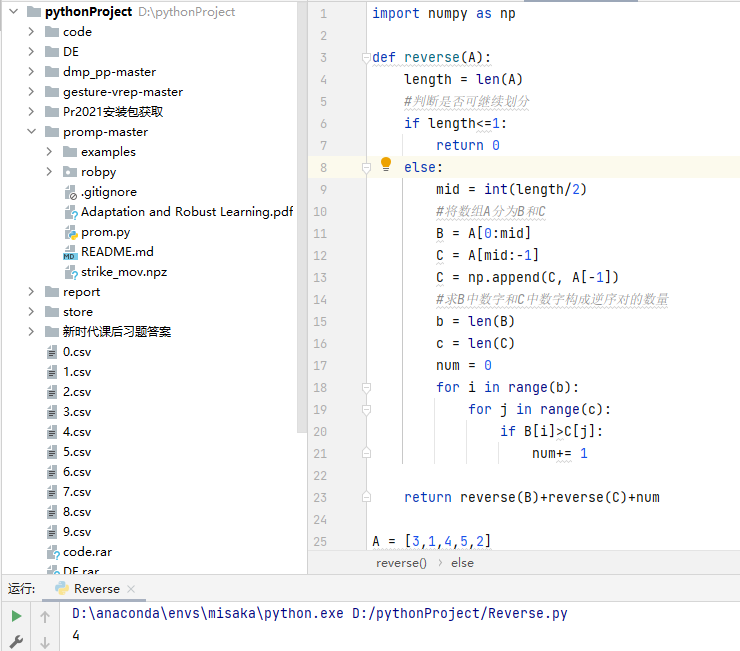
赋值：dp[i,0] = Matrix[i,0]0\le i \le col \\

dp[0,j] = Matrix[0,j] 0\le i \le row \\

输出结果

结束

代码实现及结果：



代码示例：

import numpy as np  
  
def reverse(A):  
 length = len(A)  
 *#判断是否可继续划分* if length<=1:  
 return 0  
 else:  
 mid = int(length/2)  
 *#将数组A分为B和C* B = A[0:mid]  
 C = A[mid:-1]  
 C = np.append(C, A[-1])  
 *#求B中数字和C中数字构成逆序对的数量* b = len(B)  
 c = len(C)  
 num = 0  
 for i in range(b):  
 for j in range(c):  
 if B[i]>C[j]:  
 num+= 1  
  
 return reverse(B)+reverse(C)+num  
  
A = [3,1,4,5,2]  
a = reverse(A)  
print(a)