

**实验报告**

**课程名称： 算法设计与分析**

**学 院： 应用技术学院**

**专业班级： 18计科一班**

**学 号： 201833050027**

**姓 名： 叶成宇**

1．实验目的

2．实验内容

归并排序和冒泡法

要求：程序主题部分要有注释。可演示

3．实验步骤和实验结果

一、# 归并排序

def merge\_sort(nums):

# 归并过程

def merge(left, right):

result = [] # 保存归并后的结果

i = j = 0

while i < len(left) and j < len(right): # 结束归并的条件

if left[i] <= right[j]:

result.append(left[i])

i += 1

else:

result.append(right[j])

j += 1

result = result + left[i:] + right[j:] # 剩余的元素添加到末尾

return result

# 递归过程

if len(nums) <= 1:

return nums

mid = len(nums) // 2

left = merge\_sort(nums[:mid])

right = merge\_sort(nums[mid:])

return merge(left, right)

arr\_random = [random.randrange(100) for i in range(10)] # 随机生成10个100以内的数据

print("raw\_data:")

for i in range(0, len(arr\_random)):

print("{: ^3}".format(arr\_random[i]), end=" ") # 打印原始数据

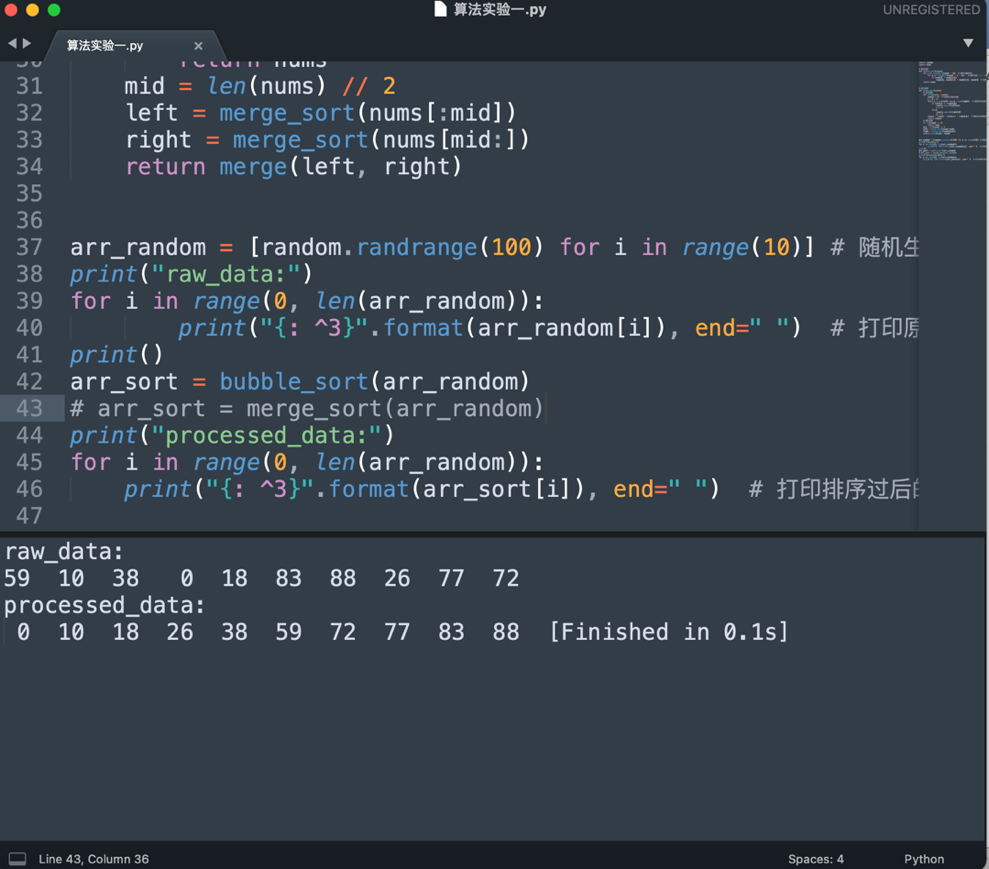
print()

arr\_sort = merge\_sort(arr\_random)

print("processed\_data:")

for i in range(0, len(arr\_random)):

print("{: ^3}".format(arr\_sort[i]), end=" ") # 打印排序过后的数据



归并排序实验结果

二、# 冒泡排序

def bubble\_sort(nums):

for i in range(len(nums) - 1): # 遍历长度减1次

for j in range(len(nums) - i - 1): # 遍历长度 - i - 1次，前面的i次已排序好

if nums[j] > nums[j+1]:

nums[j], nums[j+1] = nums[j+1], nums[j] # 如果后面的数比前面的数小，就交换两个数

return nums

arr\_random = [random.randrange(100) for i in range(10)] # 随机生成10个100以内的数据

print("raw\_data:")

for i in range(0, len(arr\_random)):

print("{: ^3}".format(arr\_random[i]), end=" ") # 打印原始数据

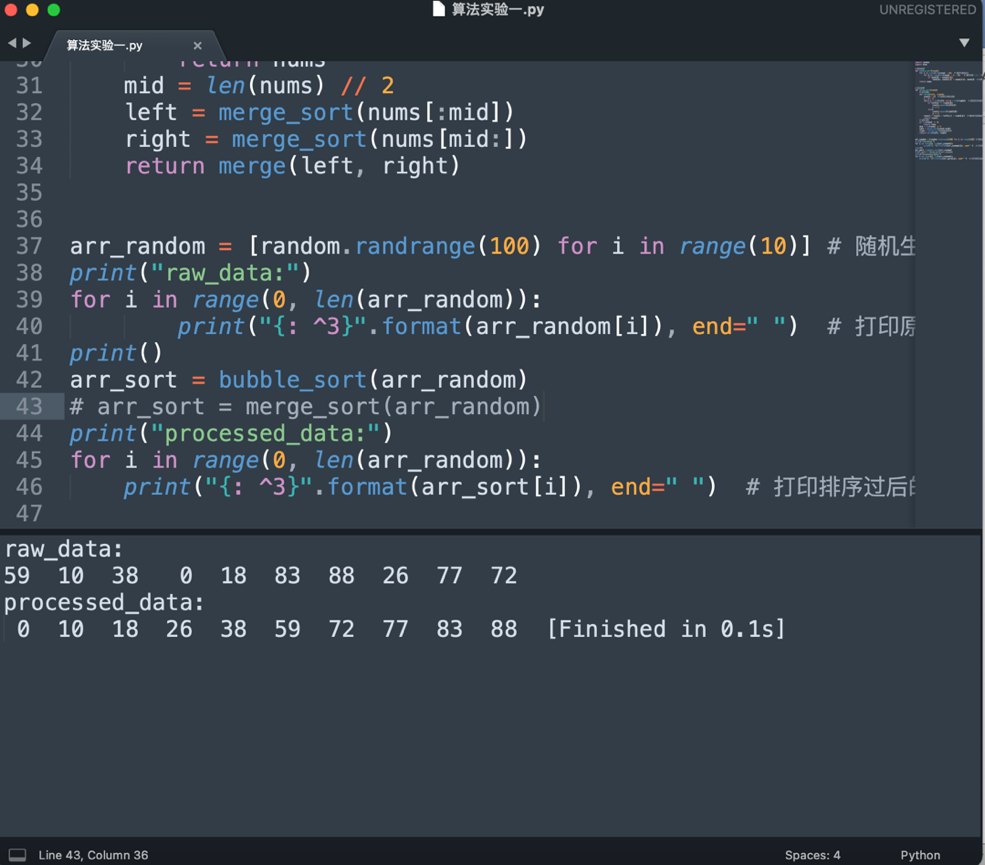
print()

arr\_sort = bubble\_sort(arr\_random)

print("processed\_data:")

for i in range(0, len(arr\_random)):

print("{: ^3}".format(arr\_sort[i]), end=" ") # 打印排序过后的数据



冒泡排序实验结果

4．分析与讨论

一、归并排序，是创建在归并操作上的一种有效的排序算法。算法是采用分治法（Divide and Conquer）的一个非常典型的应用，且各层分治递归可以同时进行。归并排序思路简单，速度仅次于快速排序，为稳定排序算法，一般用于对总体无序，但是各子项相对有序的数列。

二、冒泡排序（英语：Bubble Sort），是一种简单的排序算法。它重复地走访过要排序的数列，一次比较两个元素，如果他们的顺序错误就把他们交换过来。走访数列的工作是重复地进行直到没有再需要交换，也就是说该数列已经排序完成。这个算法的名字由来是因为越小的元素会经由交换慢慢“浮”到数列的顶端。