实验项目一：递归程序实现

实验名称：快速排序问题

**题 目**：给定n个数值，从小到大排序。

实验内容：通过快速排序问题的算法设计，进而分析算法的时间复杂性，并用编程语言实现，调试通过，得出正确答案。

实验性质：设计型

实验学时：2

实验条件：C、C++，JAVA、Python等语言，互联网络环境

**实验目的与要求**：

1. **掌握递归算法的概念和基本思想，分析并掌握快速排序算法；掌握编程语言的基本库函数；**
2. **采用三种排序（快速排序、自选一类排序算法）**

代码：

'''

name: 谭铭瑞

time:2023/10/30

project:随机生成0<=x<=10^9数组,并对其排序(快速和归并排序函数)

'''

import random

import sys

import time

def get\_shuzu(n):

# 函数功能：获取n个随机数组

shuzu = []

for i in range(0, n):

shuzu.append(random.randint(0, 10\*\*9))

return shuzu

def quick\_sort(shuzu):

# 函数功能：对数组内进行快速排序

if len(shuzu) <= 1:

# 结束的条件

return shuzu

line = shuzu[len(shuzu) // 2] # 中间值为分界

left = [x for x in shuzu if x < line]

middle = [x for x in shuzu if x == line]

right = [x for x in shuzu if x > line]

return quick\_sort(left) + quick\_sort(middle) + quick\_sort(right)

def merge\_sort(arr):

# 归并排序函数

if len(arr) <= 1:

return arr

# 将数组分为两半并递归排序

mid = len(arr) // 2

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

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

# 合并排好序的子数组

return merge(left, right)

def merge(left, right):

# 合并两个有序数组

merged = []

i = j = 0

# 比较左右两个数组的元素，按顺序合并

while i < len(left) and j < len(right):

if left[i] < right[j]:

merged.append(left[i])

i += 1

else:

merged.append(right[j])

j += 1

# 将剩余的元素添加到合并结果中

merged.extend(left[i:])

merged.extend(right[j:])

return merged

def Ptest1(shuzu\_5,shuzu\_10,shuzu\_100): # 测试快速排序法

print("排序前:\n",

shuzu\_5, "\n",

shuzu\_10, "\n",

shuzu\_100,"\n")

print("排序后:\n",

quick\_sort(shuzu\_5), "\n",

quick\_sort(shuzu\_10), "\n",

quick\_sort(shuzu\_100),"\n")

def Ptest2(shuzu\_5,shuzu\_10,shuzu\_100): # 测试归并排序法

print("排序前:\n",

shuzu\_5, "\n",

shuzu\_10, "\n",

shuzu\_100,"\n")

print("排序后:\n",

merge\_sort(shuzu\_5), "\n",

merge\_sort(shuzu\_10), "\n",

merge\_sort(shuzu\_100),"\n")

def main():

shuzu\_5 = get\_shuzu(5)

shuzu\_10 = get\_shuzu(10)

shuzu\_100 = get\_shuzu(100)

start\_time1 = time.time()

print("快速排序实际占用空间：", sys.getsizeof(Ptest1(shuzu\_5,shuzu\_10,shuzu\_100)), "字节")

end\_time1 = time.time()

print("快速排序实际执行时间：", (end\_time1 - start\_time1))

start\_time2 = time.time()

print("归并排序实际占用空间：", sys.getsizeof(Ptest2(shuzu\_5,shuzu\_10,shuzu\_100)), "字节")

end\_time2 = time.time()

print("归并排序实际执行时间：", (end\_time2 - start\_time2))

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

main()

代码运行结果