

# BTH004-归并排序

---

姓名：赵水 学号：201806150329

## 1 问题：

---

对于给定数组A，使用归并排序算法进行排序输出SA，并进行时间效率检验。

## 2 算法：

---

排序算法：

对于数组A，将其左右划分为两等分LA 和 RA，再继续对LA，RA继续划分，直到不能再继续划分后，再按照左小右大的方式进行拼接归并，得到排序后的数组SA。

时间效率的计算：

先记录算法运行前时的系统时间，再记录算法进行后的系统时间，两者之差即为算法所用的时间。同时不断增大数组容量，每个容量下进行100次实验后取均值，从而取得不同数量级下的时间效率。

## 3 代码：

---

```
import random
import time
import numpy as np
import matplotlib.pyplot as plt
from pylab import mpl
from scipy.optimize import curve_fit

mpl.rcParams['font.sans-serif'] = ['SimHei']
mpl.rcParams['axes.unicode_minus'] = False

def merge_sort(A):
    if len(A) <= 1:
        return A
    else:
        middle = len(A) // 2
        LA = A[:middle]
        RA = A[middle:]
        LA = merge_sort(LA)
        RA = merge_sort(RA)
        SA = merge(LA, RA)
        return SA

def merge(LA, RA):
    SA = []
    l = 0
    r = 0
```

```

while l < len(LA) and r < len(RA):
    if LA[l] <= RA[r]:
        SA.append(LA[l])
        l += 1
    elif LA[l] >= RA[r]:
        SA.append(RA[r])
        r += 1
SA += LA[l:]
SA += RA[r:]
return SA

def createRandomArray(numOfArray,numRange):#定义函数用来生成大小为numOfArray, 在0到
numRange范围内的整数的数组
    arr = []
    for i in range(0,numOfArray):
        arr.append(random.randint(0, numRange))
    return arr

def func(x,a,b,c):#为拟合函数确定参数
    return a * x * np.log(x+b) + c

test = [6,5,9,4,2,8,6,1,11,36,24,13,16,10,6]
print(merge_sort(test))

PowerArray = []#数据量, 用于充当图表的x轴坐标
TimeForMerge = []#在当前数据量下归并排序所用的时间, 用于充当图表的y轴坐标

power = 1 #数据量从1开始累加
while power <= 100000:
    PowerArray.append(power)#画点
    key = random.randint(0, power) #生成所需查找的数值
    TimeForMergeTemp = [] #
    count = 0
    for j in range(0,100):#每个power容量下测试一百次, 重复实验
        array = createRandomArray(power, power)

        oldtimeForMerge = time.time()
        merge_sort(array)
        newtimeForMerge = time.time()

        TimeForMergeTemp.append(newtimeForMerge-oldtimeForMerge)
        count += 1
        if count % 10 == 0:
            print("进度: " + str(power) + ": " + str(count) + "%")

    TimeForMerge.append(np.average(TimeForMergeTemp))
    power *= 10 #增大容量继续试验

plt.xlabel("样本数")
plt.ylabel("所用时间")

popt, pcov = curve_fit(func, PowerArray, TimeForMerge,maxfev = 100000)

plt.plot(PowerArray,func(np.asarray(PowerArray),*popt),'b')
plt.scatter(PowerArray,TimeForMerge)
plt.show()#绘制图表

```

## 4 图表

---

image-20201113224744472