

**《数学建模B0714160》**

**数学建模优化大作业**

**题目： 网络金融交易分配问题**

**学号：21052001**

**姓名：鲍星彤**

摘要

运用二分法和贪心算法，不断判断中值可行性，缩小解的范围，最后求解。

得出实例1的最小费用是1419.

实例2的最小费用是2199.

实例3的最小费用是5014.

实例4的最小费用是6369.

**关键词：二分法、贪心算法**

# 问题重述

网络金融，是指在国际互联网(Internet)上开展的金融业务（包括网络银行、网络证券、网络保险, 网络股票、期权等金融服务及相关内容），它不同于传统的以物理形态存在的金融活动，是存在于电子空间中的金融活动，其存在形态是虚拟化的、运行方式是网络化的。它是信息技术特别是互联网技术飞速发展的产物，是适应电子商务发展需要而产生的网络时代的金融运行模式。在此，我们探讨的网络金融交易问题场景设定如下：某一时间段内，共有m个网络支付平台（下简称平台）和n笔可能要发生的交易，每笔交易都需经平台进行，每笔交易都有其相应的交易量和交易费用（注：此项交易费用完全归处理交易的平台所有）。为了降低金融风险和维护现有m个平台的共同发展，金融监管部门依据历史数据对于每个平台都设置其最大允许交易量，即平台的最大允许交易量为，假设所有交易都必须由金融监管部门统一分配。请解答下面问题：

**问题**：全部交易完成（目标为最小化总交易费用）。假设必须要完成所有n笔交易且有成立，即每个平台的最大允许交易量都要满足。此时，交易可能会在多个平台上进行（即单笔交易可拆分），不管交易在平台上进行交易的交易量大小如何，只要交易在平台上进行了交易或者部分交易，那么平台都会收取相应的交易费用。金融监管部门希望社会总成本最小，即总交易费用最小，也就是说尽量使得单笔交易不要拆分或者尽可能少的拆分，需要为金融监管部门设置可行交易分配方案使得总交易费用最小。试建立此问题的一般数学模型并设计快速高效的求解算法，利用附件中的实例5-8进行检验，将计算结果具体信息以附件形式提交，并对算法的时间复杂度及计算结果的优劣进行分析。

# 问题分析

原本是准备用最大流最小费用模型处理的，但是这个条件不是很适合。于是改用二分法。通过不断缩小上下界求出一个合适的值。

# 模型假设

1.假设不受其他因素

# 符号说明

|  |  |
| --- | --- |
| 符号 | 说明 |
| **sum** | 所有交易的费用总和 |
| **left** | 下界 |
| **right** | 上界 |
| **mid** | 中值 |
| **money** | 剩余金额 |
|  |  |

# 模型的建立与求解

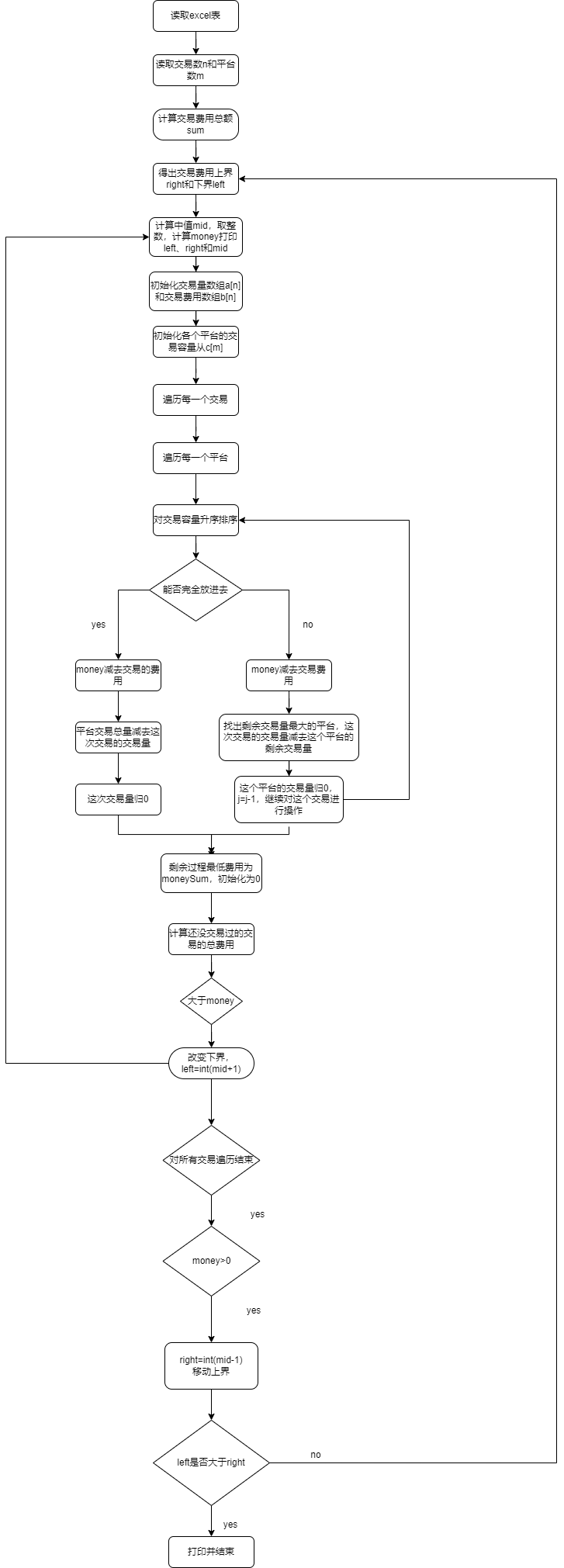
模型的思想是，不断通过二分，判断中值是否可行，缩小解的范围，最后求出解。同时运用贪心算法，从费用高的交易开始，每次在剩余交易量尽量少，但又能完全交易的平台交易，如果找不到交易平台，就先找到剩余交易量最大的进行交易，再在剩余交易量尽量少，但又能完全交易的平台交易，以此类推。

首先先对excel表进行排序，以交易费用为关键字，降序，确保交易费用高的在前，低的在后。

求得所有交易的费用总和sum。可以确定最小价格是在sum和10\*sum之间。可以给left和right赋初值，得出：。

然后可以求得mid值，。

具体流程如下：



# 模型评价与改进

### 6.1模型的优点

1. 可以计算出较为近似的值
2. 可以确定解的范围（小于等于求出来的值，大于等于所有交易的交易费用之和）
3. 速度比较快，时间复杂度为

### 6.2模型的缺点

1.不严谨，没法确定精确的解

### 6.3模型的改进

1.二分法没有问题，但是贪心算法还需要改进

# 附录

#### 对实例1：

代码：

# import ortools

# from ortools.graph import pywrapgraph

import xlrd

import xlwt

# between each pair. For instance, the arc from node 0 to node 1 has acapacity of 15 and a unit cost of 4.

a = []

# 表示每一笔交易的交易量

b = []

# 表示每一笔交易的费用

c = []

# 表示每一个平台的最大交易量

m = 10

n = 50

test = xlrd.open\_workbook('test1.xls');

sheet1\_content1 = test.sheet\_by\_index(0);

for i in range(1, n + 1):

a.append(sheet1\_content1.cell(i, 1).value)

for i in range(1, n + 1):

b.append(sheet1\_content1.cell(i, 2).value)

print(a)

print(b)

c.append(100)

c.append(200)

c.append(300)

c.append(400)

c.append(500)

c.append(500)

c.append(600)

c.append(700)

c.append(800)

c.append(900)

left = 0

for x in b:

left += x

print(left)

right = left \* 10

mid = 0

money = 0

for i in range(0, 40):

mid = (left + right) / 2

mid = int(mid)

print("left ", end='')

print(left)

print("right ", end='')

print(right)

print("mid ", end='')

print(mid)

if left>right:

print("运行结束！");

break

money = mid

a=[]

b=[]

c=[]

for j in range(1, n + 1):

a.append(sheet1\_content1.cell(j, 1).value)

for j in range(1, n + 1):

b.append(sheet1\_content1.cell(j, 2).value)

c.append(100)

c.append(200)

c.append(300)

c.append(400)

c.append(500)

c.append(500)

c.append(600)

c.append(700)

c.append(800)

c.append(900)

for j in range(0, n):

for k in range(0, m):

sorted(c)

flag = 1

if a[j] < c[k]:

money -= b[j]

c[k] -= a[j]

a[j] = 0

flag = 0

break

if a[j]>c[m-1]:

money -= b[j]

a[j] -= c[m - 1]

c[m - 1] = 0

j -= 1

break

moneySum = 0

for k in range(0, n):

if a[k] > 0:

moneySum += b[k]

if moneySum > money:

left = mid + 1

left = int(left)

break

if money > 0:

right = mid - 1

right = int(right)

运行结果：

D:\pythonProject2\venv\Scripts\python.exe D:/pythonProject2/main.py

[12.0, 180.0, 136.0, 49.0, 85.0, 15.0, 43.0, 31.0, 77.0, 111.0, 18.0, 9.0, 52.0, 190.0, 6.0, 88.0, 129.0, 53.0, 114.0, 95.0, 12.0, 166.0, 107.0, 133.0, 100.0, 9.0, 142.0, 154.0, 123.0, 97.0, 127.0, 146.0, 186.0, 153.0, 147.0, 188.0, 93.0, 183.0, 3.0, 149.0, 98.0, 195.0, 87.0, 197.0, 72.0, 160.0, 53.0, 92.0, 5.0, 130.0]

[48.0, 48.0, 47.0, 46.0, 46.0, 45.0, 44.0, 43.0, 43.0, 42.0, 42.0, 41.0, 40.0, 40.0, 38.0, 38.0, 38.0, 37.0, 37.0, 37.0, 33.0, 32.0, 32.0, 30.0, 29.0, 29.0, 29.0, 28.0, 25.0, 24.0, 22.0, 22.0, 21.0, 21.0, 21.0, 18.0, 18.0, 17.0, 17.0, 17.0, 16.0, 14.0, 12.0, 11.0, 10.0, 8.0, 8.0, 2.0, 1.0, 1.0]

1408.0

left 1408.0

right 14080.0

mid 7744

left 1408.0

right 7743

mid 4575

left 1408.0

right 4574

mid 2991

left 1408.0

right 2990

mid 2199

left 1408.0

right 2198

mid 1803

left 1408.0

right 1802

mid 1605

left 1408.0

right 1604

mid 1506

left 1408.0

right 1505

mid 1456

left 1408.0

right 1455

mid 1431

left 1408.0

right 1430

mid 1419

left 1420

right 1418

mid 1419

运行结束！

进程已结束,退出代码0

#### 对实例2

代码：

# import ortools

# from ortools.graph import pywrapgraph

import xlrd

import xlwt

# between each pair. For instance, the arc from node 0 to node 1 has acapacity of 15 and a unit cost of 4.

a = []

# 表示每一笔交易的交易量

b = []

# 表示每一笔交易的费用

c = []

# 表示每一个平台的最大交易量

m = 50

n = 50

test = xlrd.open\_workbook('test1.xls');

sheet1\_content1 = test.sheet\_by\_index(0);

for i in range(1, n + 1):

a.append(sheet1\_content1.cell(i, 1).value)

for i in range(1, n + 1):

b.append(sheet1\_content1.cell(i, 2).value)

print(a)

print(b)

c=[]

for i in range(0,m):

c.append(100)

left = 0

for x in b:

left += x

print(left)

right = left \* 10

mid = 0

money = 0

for i in range(0, 10):

mid = (left + right) / 2

mid = int(mid)

print("left ", end='')

print(left)

print("right ", end='')

print(right)

print("mid ", end='')

print(mid)

if left>right:

print("运行结束！");

break

money = mid

a=[]

b=[]

c=[]

for j in range(1, n + 1):

a.append(sheet1\_content1.cell(j, 1).value)

for j in range(1, n + 1):

b.append(sheet1\_content1.cell(j, 2).value)

for j in range(0,m):

c.append(100)

for j in range(0, n):

for k in range(0, m):

sorted(c)

flag = 1

if a[j] < c[k]:

money -= b[j]

c[k] -= a[j]

a[j] = 0

flag = 0

break

if a[j]>c[m-1]:

money -= b[j]

a[j] -= c[m - 1]

c[m - 1] = 0

j -= 1

break

moneySum = 0

for k in range(0, n):

if a[k] > 0:

moneySum += b[k]

if moneySum > money:

left = mid + 1

left = int(left)

break

if money > 0:

right = mid - 1

right = int(right)

运行结果：

[12.0, 180.0, 136.0, 49.0, 85.0, 15.0, 43.0, 31.0, 77.0, 111.0, 18.0, 9.0, 52.0, 190.0, 6.0, 88.0, 129.0, 53.0, 114.0, 95.0, 12.0, 166.0, 107.0, 133.0, 100.0, 9.0, 142.0, 154.0, 123.0, 97.0, 127.0, 146.0, 186.0, 153.0, 147.0, 188.0, 93.0, 183.0, 3.0, 149.0, 98.0, 195.0, 87.0, 197.0, 72.0, 160.0, 53.0, 92.0, 5.0, 130.0]

[48.0, 48.0, 47.0, 46.0, 46.0, 45.0, 44.0, 43.0, 43.0, 42.0, 42.0, 41.0, 40.0, 40.0, 38.0, 38.0, 38.0, 37.0, 37.0, 37.0, 33.0, 32.0, 32.0, 30.0, 29.0, 29.0, 29.0, 28.0, 25.0, 24.0, 22.0, 22.0, 21.0, 21.0, 21.0, 18.0, 18.0, 17.0, 17.0, 17.0, 16.0, 14.0, 12.0, 11.0, 10.0, 8.0, 8.0, 2.0, 1.0, 1.0]

1408.0

left 1408.0

right 14080.0

mid 7744

left 1408.0

right 7743

mid 4575

left 1408.0

right 4574

mid 2991

left 1408.0

right 2990

mid 2199

left 2200

right 2198

mid 2199

运行结束！

进程已结束,退出代码0

#### 实例3

代码：

# import ortools

# from ortools.graph import pywrapgraph

import xlrd

import xlwt

# between each pair. For instance, the arc from node 0 to node 1 has acapacity of 15 and a unit cost of 4.

a = []

# 表示每一笔交易的交易量

b = []

# 表示每一笔交易的费用

c = []

# 表示每一个平台的最大交易量

m = 50

n = 200

test = xlrd.open\_workbook('test1.xls');

sheet1\_content1 = test.sheet\_by\_index(0);

for i in range(1, n + 1):

a.append(sheet1\_content1.cell(i, 1).value)

for i in range(1, n + 1):

b.append(sheet1\_content1.cell(i, 2).value)

print(a)

print(b)

c=[]

for i in range(0,m):

c.append(416)

left = 0

for x in b:

left += x

print(left)

right = left \* 10

mid = 0

money = 0

for i in range(0, 10):

mid = (left + right) / 2

mid = int(mid)

print("left ", end='')

print(left)

print("right ", end='')

print(right)

print("mid ", end='')

print(mid)

if left>right:

print("运行结束！");

break

money = mid

a=[]

b=[]

c=[]

for j in range(1, n + 1):

a.append(sheet1\_content1.cell(j, 1).value)

for j in range(1, n + 1):

b.append(sheet1\_content1.cell(j, 2).value)

for j in range(0,m):

c.append(416)

for j in range(0, n):

for k in range(0, m):

sorted(c)

flag = 1

if a[j] < c[k]:

money -= b[j]

c[k] -= a[j]

a[j] = 0

flag = 0

break

if a[j]>c[m-1]:

money -= b[j]

a[j] -= c[m - 1]

c[m - 1] = 0

j -= 1

break

moneySum = 0

for k in range(0, n):

if a[k] > 0:

moneySum += b[k]

if moneySum > money:

left = mid + 1

left = int(left)

break

if money > 0:

right = mid - 1

right = int(right)

运行结果：

D:\pythonProject2\venv\Scripts\python.exe D:/pythonProject2/main.py

[35.0, 162.0, 119.0, 195.0, 89.0, 88.0, 163.0, 77.0, 16.0, 120.0, 71.0, 53.0, 195.0, 54.0, 191.0, 51.0, 124.0, 140.0, 172.0, 96.0, 85.0, 13.0, 3.0, 100.0, 167.0, 12.0, 125.0, 146.0, 167.0, 74.0, 113.0, 76.0, 151.0, 37.0, 171.0, 157.0, 195.0, 48.0, 10.0, 47.0, 133.0, 89.0, 26.0, 127.0, 93.0, 44.0, 87.0, 161.0, 22.0, 178.0, 115.0, 62.0, 163.0, 185.0, 177.0, 23.0, 99.0, 2.0, 129.0, 96.0, 192.0, 166.0, 78.0, 140.0, 111.0, 37.0, 34.0, 123.0, 113.0, 165.0, 65.0, 198.0, 63.0, 163.0, 150.0, 172.0, 85.0, 100.0, 103.0, 110.0, 32.0, 191.0, 102.0, 60.0, 90.0, 199.0, 77.0, 161.0, 121.0, 148.0, 30.0, 133.0, 62.0, 172.0, 156.0, 58.0, 134.0, 55.0, 180.0, 109.0, 191.0, 181.0, 37.0, 152.0, 4.0, 112.0, 123.0, 161.0, 151.0, 40.0, 49.0, 51.0, 109.0, 49.0, 30.0, 25.0, 183.0, 175.0, 16.0, 2.0, 14.0, 165.0, 142.0, 53.0, 163.0, 173.0, 108.0, 20.0, 87.0, 175.0, 148.0, 64.0, 121.0, 163.0, 48.0, 146.0, 157.0, 5.0, 164.0, 142.0, 153.0, 138.0, 26.0, 195.0, 132.0, 35.0, 88.0, 128.0, 64.0, 137.0, 129.0, 83.0, 148.0, 165.0, 171.0, 198.0, 76.0, 64.0, 11.0, 174.0, 122.0, 123.0, 175.0, 7.0, 2.0, 141.0, 97.0, 54.0, 188.0, 31.0, 172.0, 9.0, 120.0, 132.0, 163.0, 117.0, 10.0, 63.0, 75.0, 25.0, 110.0, 34.0, 87.0, 51.0, 119.0, 28.0, 140.0, 33.0, 100.0, 129.0, 178.0, 112.0, 73.0, 127.0, 108.0, 10.0, 103.0, 6.0, 199.0, 56.0]

[50.0, 50.0, 50.0, 50.0, 49.0, 49.0, 49.0, 49.0, 48.0, 48.0, 48.0, 48.0, 48.0, 48.0, 47.0, 47.0, 47.0, 47.0, 47.0, 46.0, 45.0, 45.0, 45.0, 44.0, 44.0, 44.0, 44.0, 44.0, 44.0, 44.0, 44.0, 43.0, 43.0, 42.0, 42.0, 42.0, 41.0, 41.0, 41.0, 40.0, 40.0, 40.0, 40.0, 39.0, 39.0, 39.0, 39.0, 39.0, 39.0, 39.0, 38.0, 38.0, 37.0, 37.0, 37.0, 37.0, 37.0, 36.0, 36.0, 35.0, 34.0, 34.0, 34.0, 34.0, 33.0, 33.0, 33.0, 32.0, 32.0, 32.0, 32.0, 31.0, 31.0, 31.0, 31.0, 31.0, 30.0, 29.0, 29.0, 28.0, 28.0, 28.0, 28.0, 28.0, 28.0, 27.0, 27.0, 27.0, 27.0, 27.0, 27.0, 27.0, 26.0, 26.0, 26.0, 25.0, 25.0, 25.0, 25.0, 25.0, 25.0, 25.0, 25.0, 25.0, 24.0, 24.0, 24.0, 24.0, 24.0, 23.0, 23.0, 22.0, 22.0, 21.0, 21.0, 21.0, 21.0, 21.0, 21.0, 21.0, 20.0, 19.0, 19.0, 19.0, 19.0, 18.0, 18.0, 17.0, 17.0, 16.0, 16.0, 16.0, 16.0, 16.0, 15.0, 15.0, 15.0, 14.0, 14.0, 14.0, 14.0, 14.0, 13.0, 13.0, 13.0, 12.0, 12.0, 12.0, 12.0, 12.0, 12.0, 12.0, 11.0, 11.0, 11.0, 11.0, 11.0, 11.0, 10.0, 10.0, 10.0, 10.0, 9.0, 9.0, 9.0, 8.0, 8.0, 8.0, 8.0, 7.0, 7.0, 7.0, 6.0, 6.0, 5.0, 5.0, 5.0, 5.0, 5.0, 4.0, 4.0, 4.0, 4.0, 4.0, 3.0, 3.0, 3.0, 3.0, 3.0, 3.0, 2.0, 2.0, 2.0, 2.0, 2.0, 2.0, 1.0, 1.0, 1.0, 1.0]

4972.0

left 4972.0

right 49720.0

mid 27346

left 4972.0

right 27345

mid 16158

left 4972.0

right 16157

mid 10564

left 4972.0

right 10563

mid 7767

left 4972.0

right 7766

mid 6369

left 4972.0

right 6368

mid 5670

left 4972.0

right 5669

mid 5320

left 4972.0

right 5319

mid 5145

left 4972.0

right 5144

mid 5058

left 4972.0

right 5057

mid 5014

进程已结束,退出代码0

#### 实例4

代码：

# import ortools

# from ortools.graph import pywrapgraph

import xlrd

import xlwt

# between each pair. For instance, the arc from node 0 to node 1 has acapacity of 15 and a unit cost of 4.

a = []

# 表示每一笔交易的交易量

b = []

# 表示每一笔交易的费用

c = []

# 表示每一个平台的最大交易量

m = 50

n = 200

test = xlrd.open\_workbook('test1.xls');

sheet1\_content1 = test.sheet\_by\_index(0);

for i in range(1, n + 1):

a.append(sheet1\_content1.cell(i, 1).value)

for i in range(1, n + 1):

b.append(sheet1\_content1.cell(i, 2).value)

print(a)

print(b)

c=[]

for i in range(0,m):

c.append(208)

left = 0

for x in b:

left += x

print(left)

right = left \* 10

mid = 0

money = 0

for i in range(0, 10):

mid = (left + right) / 2

mid = int(mid)

print("left ", end='')

print(left)

print("right ", end='')

print(right)

print("mid ", end='')

print(mid)

if left>right:

print("运行结束！");

break

money = mid

a=[]

b=[]

c=[]

for j in range(1, n + 1):

a.append(sheet1\_content1.cell(j, 1).value)

for j in range(1, n + 1):

b.append(sheet1\_content1.cell(j, 2).value)

for j in range(0,m):

c.append(208)

for j in range(0, n):

for k in range(0, m):

sorted(c)

flag = 1

if a[j] < c[k]:

money -= b[j]

c[k] -= a[j]

a[j] = 0

flag = 0

break

if a[j]>c[m-1]:

money -= b[j]

a[j] -= c[m - 1]

c[m - 1] = 0

j -= 1

break

moneySum = 0

for k in range(0, n):

if a[k] > 0:

moneySum += b[k]

if moneySum > money:

left = mid + 1

left = int(left)

break

if money > 0:

right = mid - 1

right = int(right)

结果：

D:\pythonProject2\venv\Scripts\python.exe D:/pythonProject2/main.py

[35.0, 162.0, 119.0, 195.0, 89.0, 88.0, 163.0, 77.0, 16.0, 120.0, 71.0, 53.0, 195.0, 54.0, 191.0, 51.0, 124.0, 140.0, 172.0, 96.0, 85.0, 13.0, 3.0, 100.0, 167.0, 12.0, 125.0, 146.0, 167.0, 74.0, 113.0, 76.0, 151.0, 37.0, 171.0, 157.0, 195.0, 48.0, 10.0, 47.0, 133.0, 89.0, 26.0, 127.0, 93.0, 44.0, 87.0, 161.0, 22.0, 178.0, 115.0, 62.0, 163.0, 185.0, 177.0, 23.0, 99.0, 2.0, 129.0, 96.0, 192.0, 166.0, 78.0, 140.0, 111.0, 37.0, 34.0, 123.0, 113.0, 165.0, 65.0, 198.0, 63.0, 163.0, 150.0, 172.0, 85.0, 100.0, 103.0, 110.0, 32.0, 191.0, 102.0, 60.0, 90.0, 199.0, 77.0, 161.0, 121.0, 148.0, 30.0, 133.0, 62.0, 172.0, 156.0, 58.0, 134.0, 55.0, 180.0, 109.0, 191.0, 181.0, 37.0, 152.0, 4.0, 112.0, 123.0, 161.0, 151.0, 40.0, 49.0, 51.0, 109.0, 49.0, 30.0, 25.0, 183.0, 175.0, 16.0, 2.0, 14.0, 165.0, 142.0, 53.0, 163.0, 173.0, 108.0, 20.0, 87.0, 175.0, 148.0, 64.0, 121.0, 163.0, 48.0, 146.0, 157.0, 5.0, 164.0, 142.0, 153.0, 138.0, 26.0, 195.0, 132.0, 35.0, 88.0, 128.0, 64.0, 137.0, 129.0, 83.0, 148.0, 165.0, 171.0, 198.0, 76.0, 64.0, 11.0, 174.0, 122.0, 123.0, 175.0, 7.0, 2.0, 141.0, 97.0, 54.0, 188.0, 31.0, 172.0, 9.0, 120.0, 132.0, 163.0, 117.0, 10.0, 63.0, 75.0, 25.0, 110.0, 34.0, 87.0, 51.0, 119.0, 28.0, 140.0, 33.0, 100.0, 129.0, 178.0, 112.0, 73.0, 127.0, 108.0, 10.0, 103.0, 6.0, 199.0, 56.0]

[50.0, 50.0, 50.0, 50.0, 49.0, 49.0, 49.0, 49.0, 48.0, 48.0, 48.0, 48.0, 48.0, 48.0, 47.0, 47.0, 47.0, 47.0, 47.0, 46.0, 45.0, 45.0, 45.0, 44.0, 44.0, 44.0, 44.0, 44.0, 44.0, 44.0, 44.0, 43.0, 43.0, 42.0, 42.0, 42.0, 41.0, 41.0, 41.0, 40.0, 40.0, 40.0, 40.0, 39.0, 39.0, 39.0, 39.0, 39.0, 39.0, 39.0, 38.0, 38.0, 37.0, 37.0, 37.0, 37.0, 37.0, 36.0, 36.0, 35.0, 34.0, 34.0, 34.0, 34.0, 33.0, 33.0, 33.0, 32.0, 32.0, 32.0, 32.0, 31.0, 31.0, 31.0, 31.0, 31.0, 30.0, 29.0, 29.0, 28.0, 28.0, 28.0, 28.0, 28.0, 28.0, 27.0, 27.0, 27.0, 27.0, 27.0, 27.0, 27.0, 26.0, 26.0, 26.0, 25.0, 25.0, 25.0, 25.0, 25.0, 25.0, 25.0, 25.0, 25.0, 24.0, 24.0, 24.0, 24.0, 24.0, 23.0, 23.0, 22.0, 22.0, 21.0, 21.0, 21.0, 21.0, 21.0, 21.0, 21.0, 20.0, 19.0, 19.0, 19.0, 19.0, 18.0, 18.0, 17.0, 17.0, 16.0, 16.0, 16.0, 16.0, 16.0, 15.0, 15.0, 15.0, 14.0, 14.0, 14.0, 14.0, 14.0, 13.0, 13.0, 13.0, 12.0, 12.0, 12.0, 12.0, 12.0, 12.0, 12.0, 11.0, 11.0, 11.0, 11.0, 11.0, 11.0, 10.0, 10.0, 10.0, 10.0, 9.0, 9.0, 9.0, 8.0, 8.0, 8.0, 8.0, 7.0, 7.0, 7.0, 6.0, 6.0, 5.0, 5.0, 5.0, 5.0, 5.0, 4.0, 4.0, 4.0, 4.0, 4.0, 3.0, 3.0, 3.0, 3.0, 3.0, 3.0, 2.0, 2.0, 2.0, 2.0, 2.0, 2.0, 1.0, 1.0, 1.0, 1.0]

4972.0

left 4972.0

right 49720.0

mid 27346

left 4972.0

right 27345

mid 16158

left 4972.0

right 16157

mid 10564

left 4972.0

right 10563

mid 7767

left 4972.0

right 7766

mid 6369

left 6370

right 6368

mid 6369

运行结束！

进程已结束,退出代码0