**暨南大学本科实验报告专用纸**

课程名称 算法分析与设计实验 成绩评定

实验项目名称 离中位数最近的K个元素 指导教师 李展

实验项目编号 实验四 实验项目类型 综合性 实验地点

学生姓名 张印祺 学号 2018051948

学院 信息科学技术 系 计算机科学 专业网络工程

实验时间 2020年3月25日下午～3月 25 日下午温度 ℃湿度

1. 问题描述

给定由n个互不相同的数组成的集合S以及正整数k，试设计一个O(n)时间算法找出S中最接近S的中位数的k个数。

1. 算法思路

1.利用lineSelect()函数，线性时间选择出集合的中位数mid。

2.遍历集合所有元素将其与mid做差并去绝对值有序存入新列表temp1。

3.利用lineSelect()函数，选择出temp1中第K+1小的数temp。

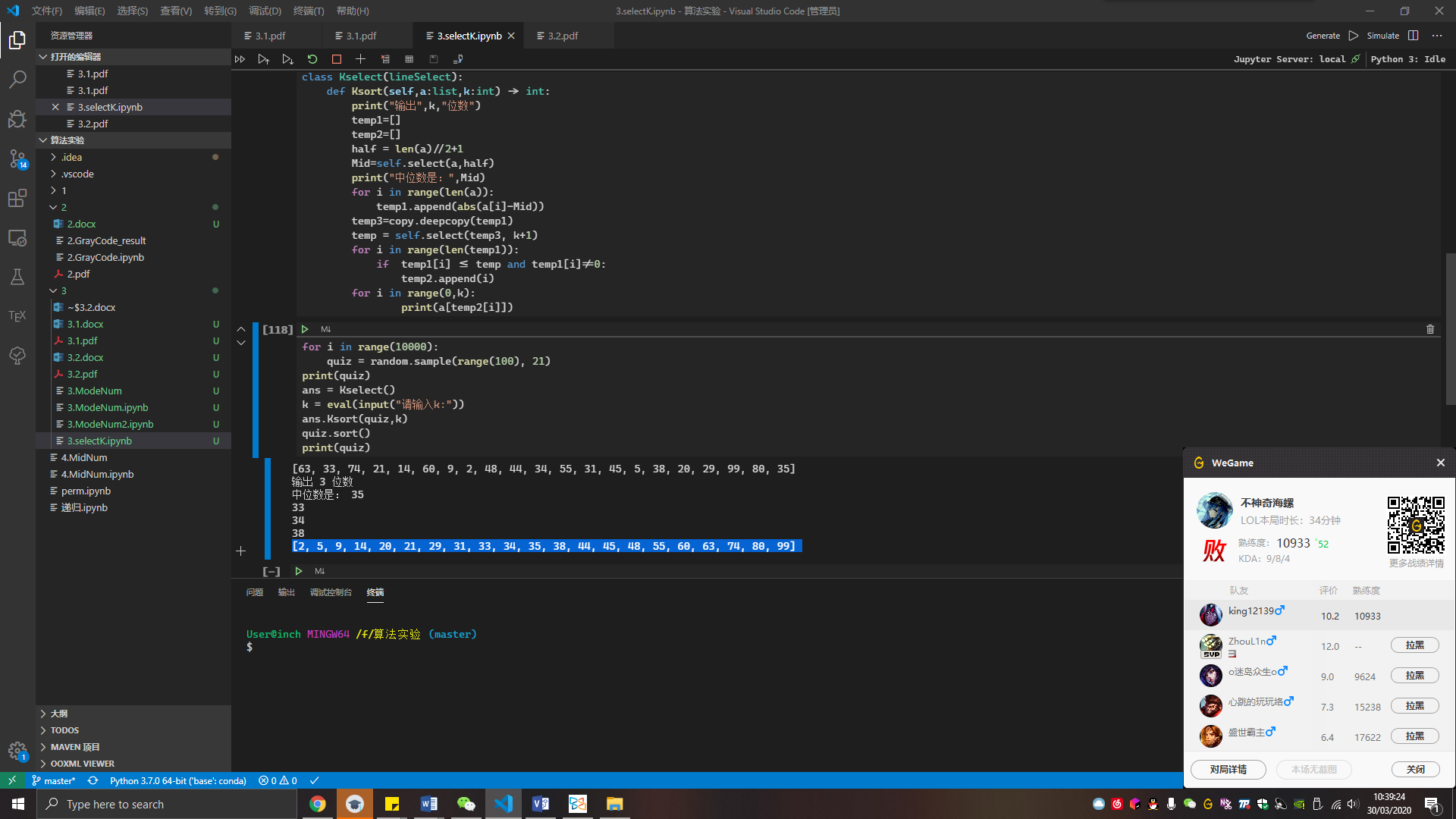
4.遍历temp1列表，将值小于temp的元素下标存入新列表temp2。

5.输出集合中下标为temp2[1：]的所有元素

1. 流程图



1. 测试结果



1. 实验总结

本算法用到了线性时间选择的原理，他能在O(n)的时间内选择出我们需要有序数组下的指定元素

假定集合元素有n位，本程序可以在O(n)的时间内找出中位数；之后进行遍历做差后取绝对值存入临时列表，时间复杂度为O(n)；然后利用线性时间选择选择出第K+1小的差值的绝对值，记录其下标到新列表，时间复杂度为O(n)；之后遍历新列表输出原列表下标为新列表的值，时间复杂度为O(k)。

故本算法的最终时间复杂度为O(n)+O(n)+O(n)+O(k)，由于，因此本算法时间复杂度，即时间复杂度为O(n)。

1. 源代码

import random

import math

import copy

class lineSelect:

    def partition(self,a:list,p:int,r:int,x:int) -> int:

        low = [m for m in a if m < x]

        high = [m for m in a if m > x]

        a[p-1:r] = low + [x] + high

        return len(low)

    def median(self, a:list) -> int:

        a.sort()

        return a[(len(a)+1)//2 - 1]

    def select(self,a:list,i:int) -> int:

        if len(a) == 1:

            return a[0]

        groups = []

        numOfGroups = int(math.ceil(len(a)\*1.0//5))

        start,end = 0,0

        for j in range(0,numOfGroups-1):

            start = j\*5

            end = start + 5

            groups.append(a[start:end])

        groups.append(a[end:])

        medians = []

        for g in groups:

            medians.append(self.median(g))

        x = self.select(medians,(len(medians)+1)/2)

        k = self.partition(a,1,len(a),x) + 1

        if k == i :

            return x

        elif k > i :

            return self.select(a[0:k],i)

        else:

            return self.select(a[k:],i-k)

class Kselect(lineSelect):

    def Ksort(self,a:list,k:int) -> int:

        print("输出",k,"位数")

        temp1=[]

        temp2=[]

        half = len(a)//2+1

        Mid=self.select(a,half)

        print("中位数是：",Mid)

        for i in range(len(a)):

            temp1.append(abs(a[i]-Mid))

        temp3=copy.deepcopy(temp1)

        temp = self.select(temp3, k+1)

        for i in range(len(temp1)):

            if  temp1[i] <= temp and temp1[i]!=0:

                temp2.append(i)

        for i in range(0,k):

                print(a[temp2[i]])

for i in range(10000):

    quiz = random.sample(range(100), 21)

print(quiz)

ans = Kselect()

k = eval(input("请输入k:"))

ans.Ksort(quiz,k)

quiz.sort()

print(quiz)