

# 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      °C 湿度     

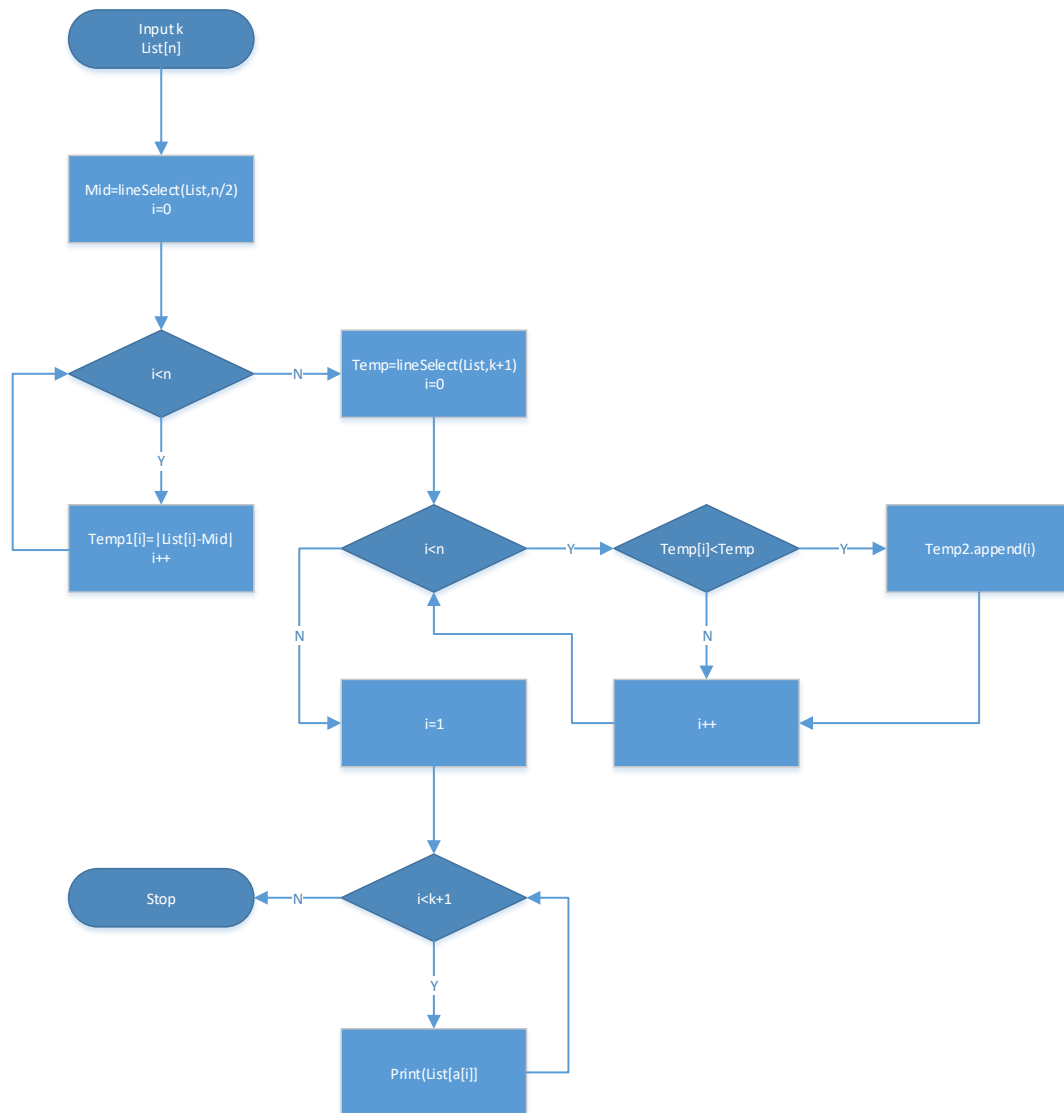
## 一、问题描述

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。

## 二、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

### 三、流程图



### 四、测试结果

```
[63, 33, 74, 21, 14, 60, 9, 2, 48, 44, 34, 55, 31, 45, 5, 38, 20, 29, 99, 80, 35]
输出 3 位数
中位数是: 35
33
34
38
[2, 5, 9, 14, 20, 21, 29, 31, 33, 34, 35, 38, 44, 45, 48, 55, 60, 63, 74, 80, 99]
```

## 五、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  $n \geq k$ ，因此本算法时间复杂度  $T(n) \leq O(4n)$ ，即时间复杂度为  $O(n)$ 。

## 六、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)

```

