

题目描述:

给定一个数组 X 和正整数 K , 请找出使表达式 $X[i] - X[i + 1] - \dots - X[i + K - 1]$ 结果最接近于数组中位数的下标 i , 如果有多个 i 满足条件, 请返回最大的 i 。

其中, 数组中位数: 长度为 N 的数组, 按照元素的值大小升序排列后, 下标为 $N/2$ 元素的值

补充说明:

1. 数组 X 的元素均为正整数;
2. X 的长度 n 取值范围: $2 \leq n \leq 1000$;
3. K 大于 0 且小于数组的大小;
4. i 的取值范围: $0 \leq i < 1000$;
5. 题目的排序数组 $X[N]$ 的中位数是 $X[N/2]$ 。

示例1

输入: [50, 50, 2, 3], 2

输出: 1

说明: 1. 中位数为50: [50, 50, 2, 3]升序排序后变成[2, 3, 50, 50], 中位数为下标4/2=2的元素50;

2. 计算结果为1: $X[50, 50, 2, 3]$ 根据题目计算 $X[i] - \dots - X[i + K - 1]$ 得出三个数0 ($X[0] - X[1] = 50 - 50$)、48 ($X[1] - X[2] = 50 - 2$) 和 -1 ($X[2] - X[3] = 2 - 3$), 其中48最接近50, 因此返回下标1。

```
1  #
2  # 语句转换
3  # @param scores int 整型一维数组 分数
4  # @param K int 整型
5  # @return int 整型
6  #
7  import math
8  class Solution:
9      def findTheStartPosition(self, scores, K):
10         # write code here
11         n = len(scores)
12         copy = scores[:]
13         copy.sort()
14         mid = copy[n//2]
15         start = 0
16         end = start+K
17         exp = scores[start]
18
19         ind = [start]
20         for i in range(start+1, end):
21             exp -= scores[i]
22             diff = abs(mid-exp)
23
24         while end < n:
25             start += 1
26             end += 1
27             exp -= scores[start-1]
28             exp += 2*scores[start]
29             exp -= scores[end-1]
30             if abs(exp-mid) < diff:
31                 ind.clear()
32                 ind.append(start)
33                 diff = abs(exp-mid)
34             elif abs(exp-mid) == diff:
35                 ind.append(start)
36         ind.sort(reverse=True)
37         return ind[0]
```