

**Author: Erin Phillips**

**Date: 10/23/2022**

# CSCI 310 - Assignment 3

---

## How it works:

This program utilizes many helper methods for achieving the desired output per the assignment instructions.

Notable functions:

```
partition(int[] arr, int m) {}
```

This function implements the Lomuto partition algorithm described in the textbook page 159. Modifications were made to ensure the algorithm works properly. Lomuto's algorithm requires the pivot be in the last position. Since the median is not consistently in the correct position, I extracted the index of the median. Converting the array to an ArrayList allows the use of `indexOf()`. Then the median is placed in the last position and the partition algorithm continues normally.

*I considered requiring arrays of the List interface, however, this would require the entire program to be modified. I found that simply having that addition where I needed was sufficient*

```
select(int[] arr, int k) {}
```

This function takes advantage of the partition function to extract important data about the subarrays S1 and S3. Allowing searches and sorting to be done *in-place* and without the use of additional arrays.

## Demonstration:

There are a total of 3 test cases, the last one is only demonstrated in the source code.

- Testing on list of size 5. This demonstrates the `insertionSort()` function.

```

-----BEGIN TESTING-----
-----TEST 1-----
TEST 1 - RANDOM ARRAY OF SIZE 5: [1, 10, 2, 11, 4]
[1] Finding the minimum value in the list (k = 1).
[1, 2, 4, 10, 11]
MIN ELEMENT FOUND: 1

[1, 10, 2, 11, 4]
[2] Finding the maximum value in the list (k = 4).
[1, 2, 4, 10, 11]
MAX ELEMENT FOUND: 11

[1, 10, 2, 11, 4]
[3] Finding the median value in the list (k = 2).
[1, 2, 4, 10, 11]
MEDIAN ELEMENT FOUND: 4

```

---

- Testing on list of size 15. Each test finds the min, max, and median of the array.
- The array was re-initialized before each sub-test.
- You will see the original array, the divisions of the array into sequences of 5 elements, the computation of the median, and the results after the partition function.
- 1st element (min):

```

-----TEST 2-----
TEST 2 - RANDOM ARRAY OF SIZE 15: [100, 99, 98, 97, 96, 95, 94, 93, 92, 91, 90, 89, 88, 87, 86]
[1] Finding the minimum value in the list (k = 1).
Partition from index 0 to 4: [96, 97, 98, 99, 100]
Partition from index 5 to 9: [91, 92, 93, 94, 95]
Partition from index 10 to 14: [86, 87, 88, 89, 90]
Array of medians:
[88, 93, 98]
After partition around 93: [91, 92, 90, 86, 87, 88, 89, 93, 94, 95, 99, 100, 96, 97, 98]
Size of S1: 7. Size of S2: 7
Subarray to be searched:
[91, 92, 90, 86, 87, 88, 89]
Partition from index 0 to 4: [86, 87, 90, 91, 92]
Partition from index 5 to 6: [88, 89]
Array of medians:
[88, 90]
After partition around 90: [86, 87, 89, 88, 90, 91, 92]
Size of S1: 4. Size of S2: 2
Subarray to be searched:
[86, 87, 89, 88]
[86, 87, 88, 89]
MIN ELEMENT FOUND: 86

```

- nth element (max):

```
[100, 99, 98, 97, 96, 95, 94, 93, 92, 91, 90, 89, 88, 87, 86]
[2] Finding the maximum value in the list (k = 14).
Partition from index 0 to 4: [96, 97, 98, 99, 100]
Partition from index 5 to 9: [91, 92, 93, 94, 95]
Partition from index 10 to 14: [86, 87, 88, 89, 90]
Array of medians:
[88, 93, 98]
After partition around 93: [91, 92, 90, 86, 87, 88, 89, 93, 94, 95, 99, 100, 96, 97, 98]
Size of S1: 7. Size of S2: 7
Subarray to be searched:
[94, 95, 99, 100, 96, 97, 98]
Partition from index 0 to 4: [94, 95, 96, 99, 100]
Partition from index 5 to 6: [97, 98]
Array of medians:
[96, 97]
After partition around 97: [94, 95, 96, 97, 100, 98, 99]
Size of S1: 3. Size of S2: 3
Subarray to be searched:
[100, 98, 99]
[98, 99, 100]
MAX ELEMENT FOUND: 100
```

---

- $n/2$  element (median):

```
[100, 99, 98, 97, 96, 95, 94, 93, 92, 91, 90, 89, 88, 87, 86]
[3] Finding the median value in the list (k = 7).
Partition from index 0 to 4: [96, 97, 98, 99, 100]
Partition from index 5 to 9: [91, 92, 93, 94, 95]
Partition from index 10 to 14: [86, 87, 88, 89, 90]
Array of medians:
[88, 93, 98]
After partition around 93: [91, 92, 90, 86, 87, 88, 89, 93, 94, 95, 99, 100, 96, 97, 98]
Size of S1: 7. Size of S2: 7
MEDIAN ELEMENT FOUND: 93
```