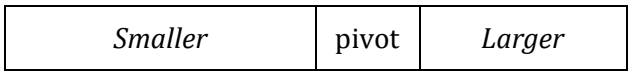
CSC 530/730
Homework 4
(100 points)
Due Date: Sunday, September 20

Problems

Statisticians often are interested in the *median* value in a collection of data. In a collection, about the same number of values are no less than the median value as are no greater than the median value. One way to find the median is to sort the data and take the value that is at - or nearly at - the center of the collection. But sorting does more than necessary to find the median. You need to find only the *k*-th smallest entry in the collection for an appropriate value of *k*. To find the median of *n* items, you would take *k* as *n/2* rounded up – that is, the smallest integer greater than or equal to *n/2*.

You can use the partitioning strategy of quick sort to find the *k*-th smallest element in an array. After finding the pivot and forming the subarrays *Smaller* and *Larger*, as shown in the following figure, you can draw one of the following conclusions:

- If *Smaller* contains *k* or more elements, it must contain the *k*-th smallest element.
- If *Smaller* contains *k-1* elements, the *k*-th smallest element is the pivot.
- If *Smaller* contains fewer than *k-1* element, the *k*-th smallest element is in *Large*.



1) In this project, you will implement a ***recursive*** function to find the *k*-th smallest element in an *unsorted* array, and then use your function to find the median in the array.

   Hint: Think about how to modify the recursive function `partition(int left, int right)` in the example program *Example14AdvancedSort* and transform it to a recursive function to find the *k*-th smallest element in an unsorted array.

2) Write necessary statements in `main()` function to either allow users to enter a collection of non-negative integers or let the program randomly generate non-negative integers, display them, and find the median. Here are some examples when running the Homework 4 project.

```
Select from:
1. Read Array
2. Generate Array
3. Print Array
4. Median
0. Quit
1
Enter elements (negative to end): 4 1 5 8 2 9 8 7 -1

Select from:
1. Read Array
```

```
2. Generate Array
3. Print Array
4. Median
0. Quit
3
4 1 5 8 2 9 8 7

Select from:
1. Read Array
2. Generate Array
3. Print Array
4. Median
0. Quit
4
Median: 5

Select from:
1. Read Array
2. Generate Array
3. Print Array
4. Median
0. Quit
2

Select from:
1. Read Array
2. Generate Array
3. Print Array
4. Median
0. Quit
3
13 14 2 8 14 12 0 11 5

Select from:
1. Read Array
2. Generate Array
3. Print Array
4. Median
0. Quit
4
Median: 11

Select from:
1. Read Array
2. Generate Array
3. Print Array
4. Median
0. Quit
0

Thanks for using my program.
```

Submission

Compress the JAVA project folder into a *.zip* file and submit it on Blackboard.