Problem Set 1, Programming Part
Quickselect with the Median-of-Medians Pivot

Due: 11:55pm Thursday, October 3, 2019

1 Programming Assignment

The assignment is to design and implement Quickselect with the median-of-medians pivot, where the medians are of groups of size 9.

Input:	An array A of n distinct nonnegative integers and an integer k in the set $\{1, 2, \ldots, n\}$.
Output:	The k^{th} smallest element of A.

A Java template has been provided containing an empty method `Quickselect`, which takes an integer array A and an integer k as arguments, and returns the k^{th} smallest element of A. Your task is to write the body of the `Quickselect` method. In case of incorrect inputs, you should return -1.

You must use the provided Java template as the basis of your submission. You may not change the name, return type, or parameters of the `Quickselect` method or the class. The `main` method in the template contains code to help you test your implementation by entering test data or reading it from a file. You may modify the `main` method because your submission will be tested using a different `main` method. You can use any helper methods or any helper classes. You can use any built-in class or write your own classes and data structures. We advise you to put all the classes you write in the same file, but no other class except the provided one should be declared as a public class.

2 Examples

The table below shows the correct output of the `Quickselect` method on various test inputs.

Input Array A	k	**Output**
10, 35, 12, 243	2	12
1, 24, 3, 70	1	1
85, 100, 99, 50	5	-1
85	1	85

3 Grading

The programming assignment will be marked out of 40, based on a combination of automated testing (using large test arrays similar to the ones posted on conneX) and human inspection.

There are several possible implementations for Quickselect. For an input array containing n values, the optimal implementation is $O(n)$. The mark for each submission will be based on both the asymptotic worst case running time and the ability of the algorithm to handle inputs of different sizes. The table below shows the expectations associated with different scores.

Score	Description
0 - 15	Submission does not compile or does not conform to the provided template.
16 - 30	The implemented algorithm is not $O(n)$ or is substantially inaccurate on the tested inputs.
31 - 40	The implemented algorithm is $O(n)$ and gives the correct answer on all tested inputs.

To be properly tested, every submission must compile correctly as submitted, and must be based on the provided template. If your submission does not compile for any reason (even trivial mistakes like typos), or was not based on the template, it will receive at most 15 out of 40. The best way to make sure your submission is correct is to download it from conneX after submitting and test it. You are not permitted to revise your submission after the due date, and late submissions will not be accepted, so you should ensure that you have submitted the correct version of your code before the due date. conneX will allow you to change your submission before the due date if you notice a mistake. After submitting your assignment, conneX will automatically send you a confirmation email. If you do not receive such an email, your submission was not received. If you have problems with the submission process, send an email to the instructor before the due date.