

CSC 226 - SPRING 2017
ALGORITHMS AND DATA STRUCTURES II
PROGRAMMING ASSIGNMENT 1
UNIVERSITY OF VICTORIA

1 Programming Assignment

An army battalion has n soldiers and the commander of the battalion would like to arrange a boxing tournament. There will be two categories, lightweight and heavyweight. The commander also wants to have a somewhat equal number of contestants in each category. Therefore, he needs to find the soldier with the median weight, then everyone below or equal to that weight will fall into the lightweight category and the remaining soldiers will be in the heavyweight category. Your job is to find an algorithm that will help him select the median weight soldier in the shortest possible time, because efficiency is very important in the army. We can assume that the weight of the soldiers are rounded up to the nearest integer.

Input: An array A of n non-negative integers.

Output: The $\frac{n}{2}^{th}$ integer.

A Java template has been provided containing an empty function `LinearSelect`, which takes an integer array A and an integer k as argument, and returns the k^{th} integer in A when sorted in an ascending order. Your task is to write the body of the `LinearSelect` function. You may assume that the input array A will always conform to the specification above (containing no negative values). Your code is not required to check for incorrectly formed input data.

You must use the provided Java template as the basis of your submission, and put your implementation inside the `LinearSelect` function in the template. You may not change the name, return type or parameters of the `LinearSelect` function. The main function 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 function, because your submission will be tested using a different main function. Only the contents of the `LinearSelect` function and associated helper functions (if any) will be marked.

2 Examples

The table below shows the correct output of the `LinearSelect` function on various test inputs.

Input Array	Median
3, 75, 12, 20	12
1, 2, 4, 5, 3	3
75	75
null	-1

3 Evaluation Criteria

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

There are several possible implementations for LINEARSELECT. 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.
15 - 30	The implemented algorithm is not $O(n)$ or is substantially inaccurate on the tested inputs.
30 - 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.