

Observations on the timing measurements:

-rearrangeParticipants has Big- $O(n^2)$  time complexity. Run time grows quadratically with  $n$  increasing.

-displaySeniorsIncreasingOrder has Big- $O(n^2)$  time complexity. Run time grows quadratically with  $n$  increasing.

-displayNonSeniorsIncreasingOrder has Big- $O(n)$  time complexity. Run time grows linearly with  $n$  increasing. If we double  $n$ , run time increases doubly as well.

-displayIncreasingOrder has Big- $O(n)$  time complexity. Run time grows linearly with  $n$  increasing. If we double  $n$ , run time increases doubly as well.

- rearrangeParticipants function is not linear, because it has Big- $O(n^2)$  time complexity and will grow up quadratically depending what  $n$ -size is.
- algorithm does not use tail recursion. To be tail recursive, recursive call of this function has to be the last statement. It is not possible within the restrictions, since we need to return integer as a counter of seniors (1+recursive call).