Emilia Mioduszewska
Piotr Węglewski

Task 1

Computational complexity estimation and algorithm profiling

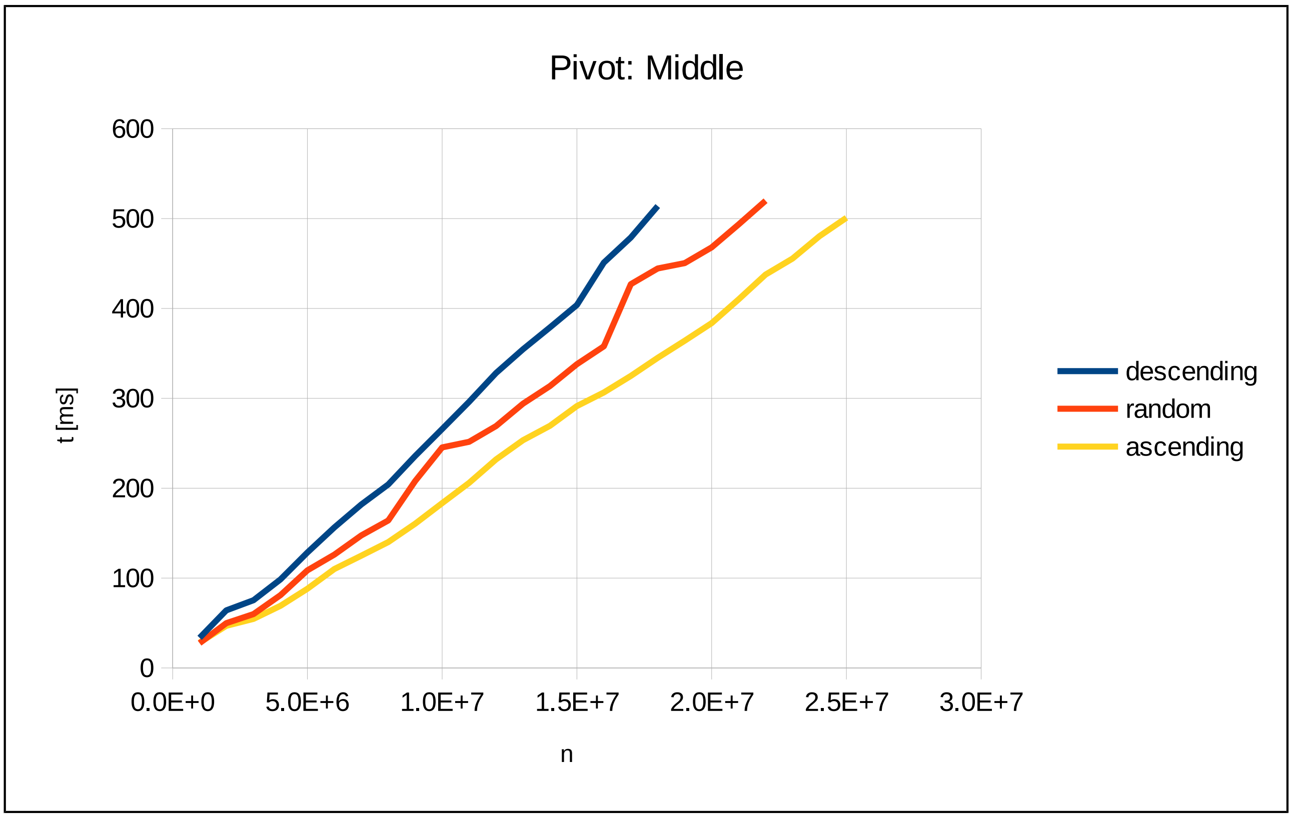
1. Experimental Procedure and expected results.

   We decided to implement a quick sort algorithm in Java language. Profiling is based on time of function execution with 1ms accuracy. In our procedure we sort the array of integers. There are 3 possibilities to initialize the array: random numbers, numbers ordered ascending and numbers ordered descending. The array of the same size is sorted 10-times and a median of all times is calculated to avoid the aberration. The size of the array is increased with a resolution until the time needed to sort it exceeds 500ms. The resolution was different for each test, but it was always set to achieve at least 20 measurement for every array size.

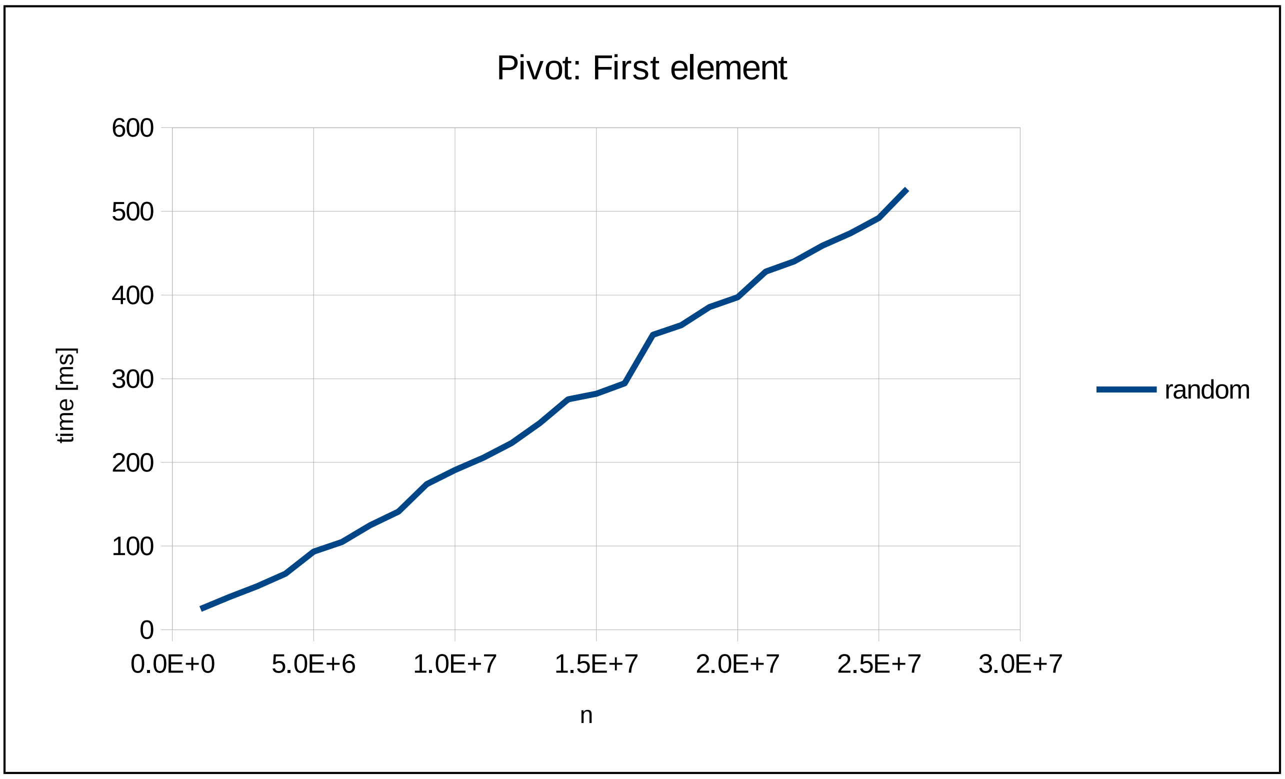
   The quick sort algorithm is expected to demonstrate the logarithmic complexity in best case scenario and n^2 complexity in worst case scenario.

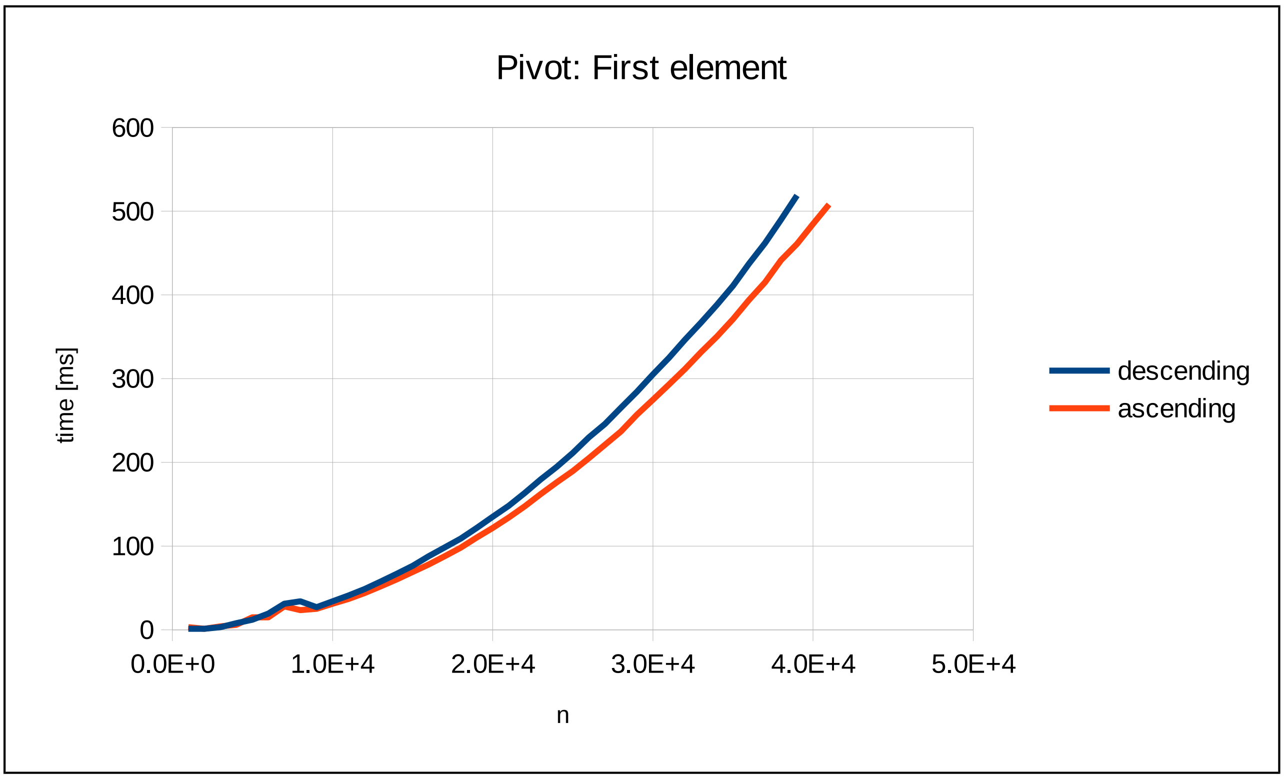
2. Experiment results.

   Middle element as dividing one.

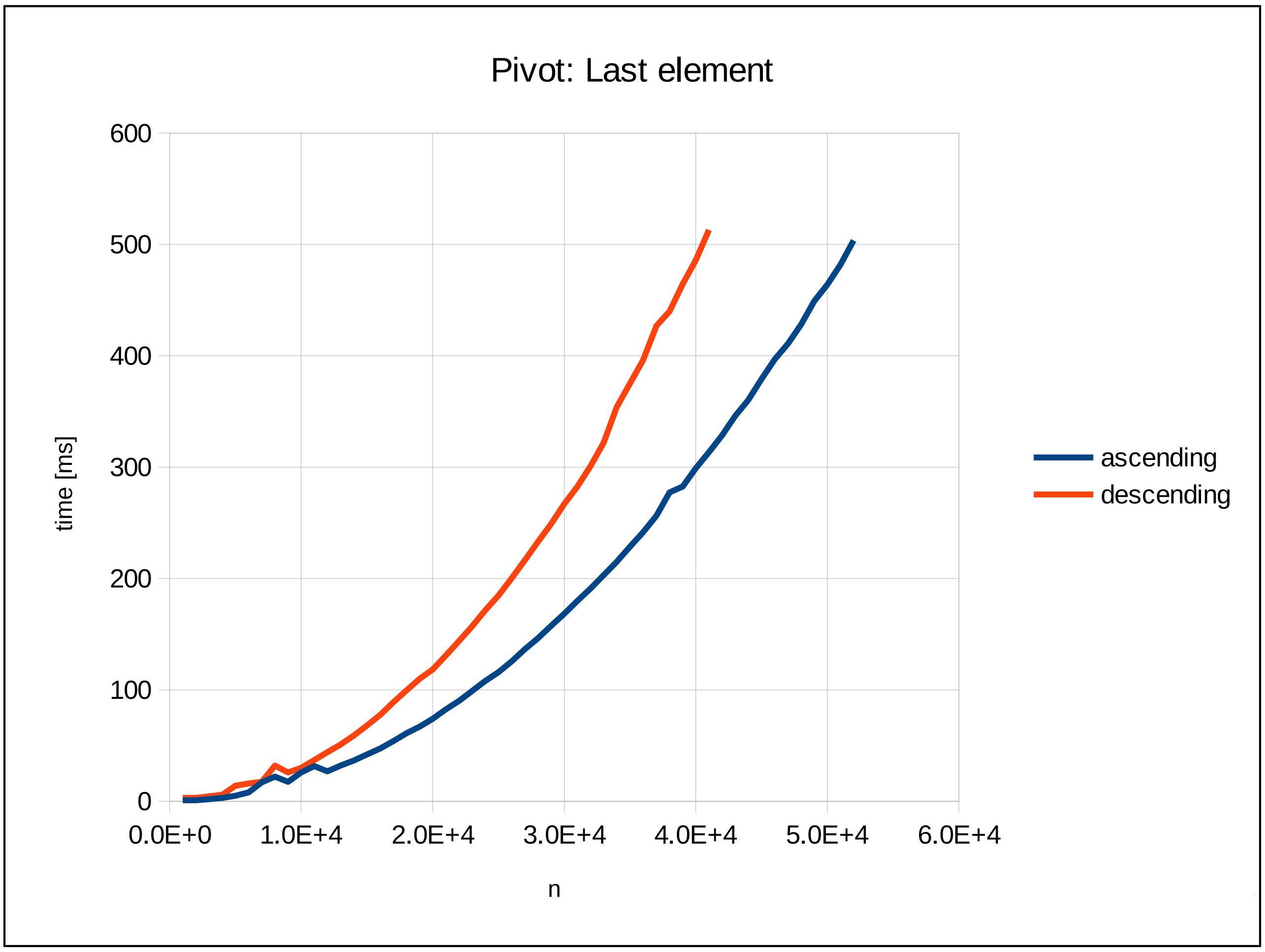


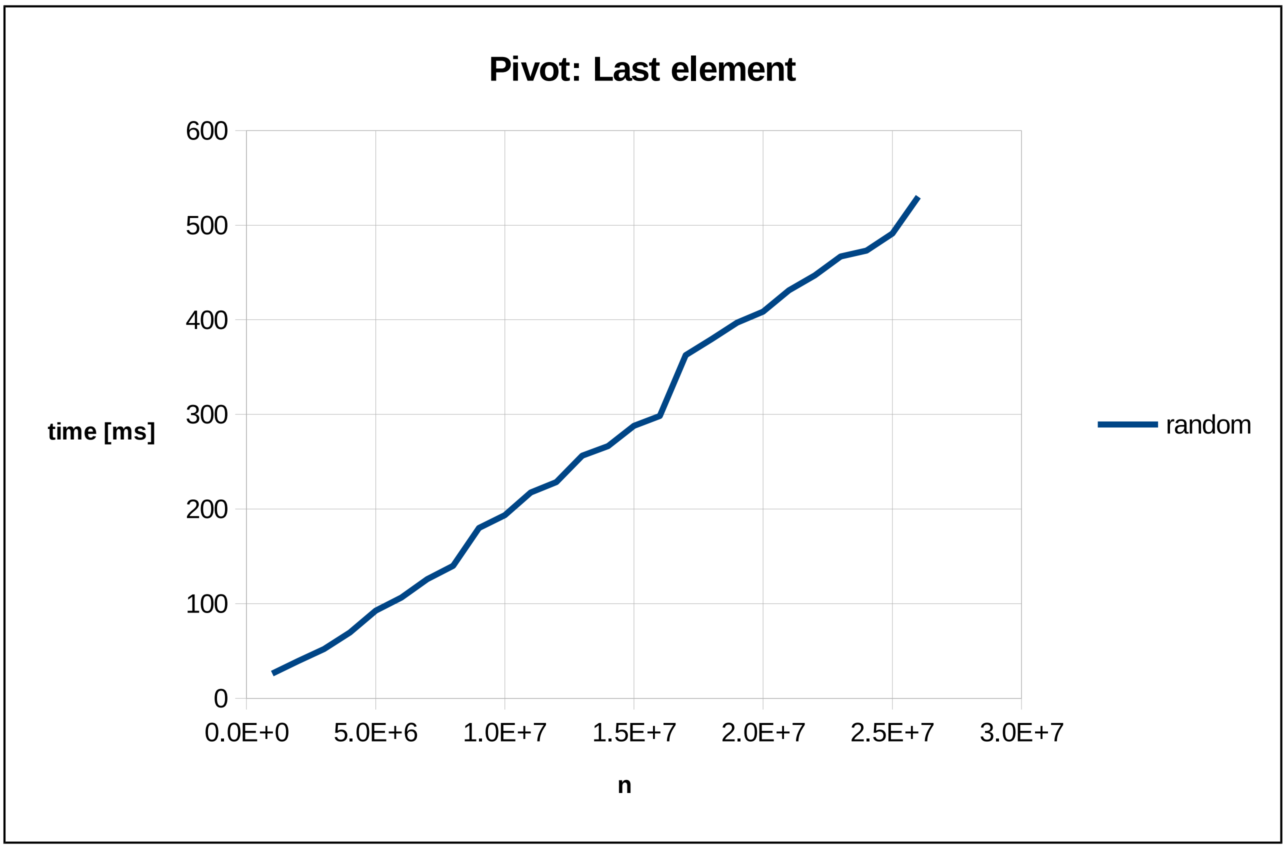
First element as dividing one.





Last element as dividing one.





3. Discussion of achieved results

Results of our experiments covers the initial expectations. The wort case scenario graph appears to be a part of n^2 function and respectively best case scenario graph is similar to part of nlogn function. The worst case scenario did not appear in the case with middle element pivot. All the test cases show the difference in algorithm complexity when elements has to be exchanged (ordered descending) and the already sorted array (ordered ascending). The graphs for random elements has some abbreviations, because of the procedure that generates different random array for every size considered in experiment. Our experiment also shows how huge is the difference in complexity for worst case and best case scenario. This confirms that this algorithm is very sensitive to incorrect implementation, because of the pivot, which position is crucial in case of quick sort.