Assignment No. 3: Analysis & Comparison of Advanced Sorting Methods – Heapsort and Quicksort

Allocated time: 2 hours

Implementation

You are required to implement **correctly** and **efficiently** the *Heapsort* and *Quicksort* advanced sorting methods.

You may find any necessary information and pseudo-code in your course notes, or in the book[1]:

- *Heapsort*: chapter 6 (Heapsort)
- *Quicksort*: chapter 7 (Quicksort)

Evaluation

! Before you start to work on the algorithms evaluation code, make sure you have a correct algorithm! You will have to prove your algorithm(s) work on a small-sized input.

1. You are required to compare the two sorting procedures in the **average** case. Remember that for the **average** case you have to repeat the measurements m times (m=5) and report their average; also for the **average** case, make sure you always use the **same** input sequence for the two methods – to make the comparison fair.

2. This is how the analysis should be performed:
   - vary the dimension of the input array (*n*) between [100…10000], with an increment of maximum 500 (we suggest 100).
   - for each dimension, generate the appropriate input sequence for the method; run the method, counting the operations (assignments and comparisons, may be counted together).

   ! Only the assignments and comparisons performed on the input structure and its corresponding auxiliary variables matter.

3. Generate a chart which compares the two methods under the total number of operations, in the **average** case. If one of the curves cannot be visualized correctly because the other has a larger growth rate, place that curve on a separate chart as well. Name your chart and the curves on it appropriately.

4. Interpret the charts and write your observations in the header (block comments) section at the beginning of your main .cpp file.

[1] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein. *Introduction to Algorithms*

5. Evaluate Quicksort in the **best** and **worst** cases also – total number of operations. Compare the performance of Quicksort in the three analysis cases. Interpret the results.