PWr Week 5

Algorithms and Data Structures

Laboratory – **List 5**

1. Implement function `void showArray(int array[], int size)` which present the array in one line: after every value put a comma.
2. Implement function `void insertSort(int array[], int size)` which sort (using insertsort) in increasing order in which the sorted part of array grows from right side (starting from the higher index). Print state of the array after each execution of outer loop.
3. Implement function `void bubbleSort(int array[], int size)` which sort (using bubble sort) in increasing order in which the sorted part of array grows from left side (starting from the zero's index). Print state of the array after each execution of outer loop.
4. Implement function `void mergeSort(int array[], int size)` for sorting in increasing order. You can use skeleton from a lecture. For this task a test only check the final state of array.
5. Compare times of the sorting algorithms from tasks 1-3 for 100, 10.000, 100.000 numbers. Is it consistent with the theory? Write your own another `main()` function to do that.

For **10 points** present solutions for this list till **Week 6**.
For 8 **points** present solutions for this list till **Week 7**.
For 5 **points** present solutions for this list till **Week 8**.
After Week 8 the list is closed.

Appendix 1

The solution of task 1,2,3,4 will be automated tested with tests from console of presented below format.

If a line starts from '`#`' sign, the line have to be ignored.

If a line has a format:
`IS` *`n`*
in next lines there will be *n* integer numbers separated by space or newline. The array have to be sorted using `insertSort` function, writing on the console state of array after every big step of the algorithm. Before start the function write in one line "insertSort" and initial state of the array in next line.

If a line has a format:
`BS` *`n`*
in next lines there will be *n* integer numbers separated by space or newline. The array have to be sorted using `bubbleSort` function, writing on the console state of array after every big step of the algorithm. Before start the function write in one line "bubbleSort" and initial state of the array in next line.
If a line has a format:

`MS` *`n`*
in next lines there will be *n* integer numbers separated by space or newline. The array have to be sorted using `mergeSort` function, writing on the console final (sorted) state of array. Before start the function write in one line "mergeSort" and initial state of the array in next line.

If a line has a format:
`HA`
your program has to end the execution, writing as the last line "END OF EXECUTION". Every test ends with this line.
For example for input test:
```
IS 5
6 9 4 7 5
BS 4
3 9 6 8
MS 4
3 9 6 8
HA
```
The output have to be:
```
insertSort
6,9,4,7,5,
6,9,4,5,7,
6,9,4,5,7,
6,4,5,7,9,
4,5,6,7,9,
bubbleSort
3,9,6,8,
3,6,9,8,
3,6,8,9,
3,6,8,9,
mergeSort
3,9,6,8,
3,6,8,9,
END OF EXECUTION
```