Algorithm & complexity

Lab: sort stability 2

# Checking the stability of algorithms (implementation without changes)

Check the stability of the algorithms (and their implementation from the SortStability2 task on GitHub): **merge sort**, **quick sort**, **Shell sort**.

Modify the test data in the SortStabilityApp application so that after sorting it it is possible to determine whether the given sorting algorithm is stable.

**Enter the test data in the field below (copy from the source code):**

|  |
| --- |
| items.add(new Item(10, 0)); items.add(new Item(5, 1)); items.add(new Item(10, 2)); items.add(new Item(2, 3)); items.add(new Item(3, 4)); items.add(new Item(1, 5)); |

Complete the table below with the test results for the unchanged implementation of the algorithms.

|  |  |
| --- | --- |
| **Algorith** | **Stable (Yes/No)** |
| Merge sort | No |
| Quick sort | NO |
| Shell sort | NO |

# Modification of the implementation of algorithms

Modify the implementation of unstable algorithms (only those for which it is possible) to make them stable. Make changes to the SimpleSort code repository on GitHub (this will be assessed).

# Checking the stability of algorithms (implementation after changes)

Check the stability of the sorting algorithms after the changes made and fill in the table below.

|  |  |
| --- | --- |
| **Algorith** | **Stable (Yes/No)** |
| Merge sort | Yes |
| Quick sort | Yes |
| Shell sort | Yes |

# Description of the introduced changes

Describe the changes made in the source code:

* Changing the implementation of which algorithm made it stable?
* What was this change about?
* What logical operation was changed (eg select an item, replace items …)?

|  |
| --- |
| during comperasiton i used >= instead of > and this is make stable. |