The logo of TAMK UAS. 

Searching and sorting test-bed

by Dmitrii Bacherikov

**Contact information**

Email: [dmitrii.bacherikov@tuni.fi](mailto:dmitrii.bacherikov@tuni.fi)

Tel.: 046-611-2164

Target points: 5

Compilation and run information, screenshots from Terminal

Graphical user interface, text, application

Description automatically generated

General design:

I implemented 1.1-1.4 parts of the project with the common menu and endless implementation until the user enter ‘5’ to quit. There is fool-tolerance feature implemented.

Text

Description automatically generated

P1 version:

Text

Description automatically generated

The list containing 100,000 odd sequential entries is generated. Then we assign odd and even targets (in the middle of the list). We can see that unsuccessful search takes twice more time since it fulfils twice more comparisons. Results look reliable.

**P2 version:**

Text

Description automatically generated

The same list containing 100,000 odd sequential entries is generated. Then here we assign RANDOMLY odd targets to be searched (100 random numbers). We fulfill 100 searches using SEQUENTIAL and BINARY search methods. It’s quite obvious that binary search takes significantly less time than sequential search. Results look reliable and expected.

**P3 version:**

Text

Description automatically generated

In this part I decided to implement INSERTION sort method using both ‘ordered\_list’ and ‘sortable\_list’. It was quite interesting to compare time spent on implementation of each method. ‘sortable\_list:insertion\_sort()’ method took almost twice less time than ‘ordered\_list:insert()’ method. The user can select how many positions will be displayed in the outcome.

P4 version:

Text

Description automatically generated

In this part I implemented merge sort method. Its time performance is a lot better. We can see it in the output.

P5 version:

Text

Description automatically generated

**Hourly report**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Year** | **Month** | **Day** | **Workhours** | **Work tasks** |
| 2021 | November | 28 | 2 | Reading chapter 7 in the textbook and researching the project requirements |
| 2021 | November | 29 | 2 | Creating the project, joining the source code from the textbook |
| 2021 | November | 30 | 4 | Adding missing pieces for part 1 |
| 2021 | December | 1 | 1 | Testing sequential search implemenation |
| 2021 | December | 2 | 4 | Debugging part 1, implementation of part 2 |
| 2021 | December | 3 | 2 | Testing of part 2, checking performance of 2 methods |
| 2021 | December | 4 | 2 | Implementation of part 3 and researching part 4 |
| 2021 | December | 5 | 1 | Making report |
| 2021 | December | 6 | 4 | Implementation of part 4, testing all parts and making the final report |
| 2021 | December | 10 | 3 | Implementation of part 5 |
|  |  |  |  |  |
|  |  |  |  |  |
| **WORK HOURS TOGETHER** | | | 25 |  |