




Work on project. Stage 2/4: Jumping bubbles

47 users solved this problem. Latest completion was 12 days ago.

Project: [Phone Book](#)

 Hard  13 minutes 

§1. Description

It's ineffective, isn't it? You have to iterate over every element of the numbers list every time you want to find someone's number. But this is the only way if your list contains unordered data. Any number can be anywhere in the list, so you are forced to check every element.

In this stage, you should sort the list of numbers alphabetically by the owner's name. Sort the list using the bubble sort algorithm and search in the list using the jump search algorithm.

After sorting, search for 500 phone numbers using the same list from the previous stage. Note how long it takes to sort the list and also measure the time the program spent during the searching. Don't include the sorting time into the searching time, because the list of numbers stays sorted after every search request. If you want to save the sorted list into the file so you don't have to sort it again, then do not override the file with an unsorted list of phone numbers. This file will also be required in the next stage.

If sorting takes a very long time (more than 10 times longer than all 500 iterations of the linear search), you should stop sorting and use the linear search. Look at the second example to see what you need to output.

§2. Example

Output both approaches one after another and see which one is faster. Output example is shown below. Note that you can get totally different sorting and searching times!

Example 1:

```
Start searching (linear search)...
Found 500 / 500 entries. Time taken: 1 min. 56 sec. 328 ms.

Start searching (bubble sort + jump search)...
Found 500 / 500 entries. Time taken: 9 min. 15 sec. 291 ms.
Sorting time: 8 min. 45 sec. 251 ms.
Searching time: 0 min. 30 sec. 40 ms.
```

Example 2:

```
Start searching (linear search)...
Found 500 / 500 entries. Time taken: 2 min. 01 sec. 134 ms.

Start searching (bubble sort + jump search)...
Found 500 / 500 entries. Time taken: 22 min. 14 sec. 482 ms.
Sorting time: 20 min. 12 sec. 251 ms. – STOPPED, moved to linear search
Searching time: 2 min. 02 sec. 231 ms.
```

[Code Editor](#) [IDE](#)



- ✓ IDE is responding IntelliJ IDEA 2019.3
- ✓ Plugin is responding 3.2-2019.3-3686

7 / 7 Prerequisites

- ✓ [Jump search](#) Stage 2/4
- ✓ [The sorting problem](#) Stage 2/4
- ✓ [Bubble sort](#) Stage 2/4
- ✓ [Jump search in Java](#) Stage 2/4
- ✓ [Bubble sort in Java](#) Stage 2/4

Show all

This content was created 11 months ago and updated 1 day ago. [Share your feedback below in comments to help us improve it!](#)

- [Comments \(3\)](#)
- [Hints \(0\)](#)
- [Useful links \(0\)](#)
- [Solutions \(0\)](#)

Share something, Sergey Kubatko

Sort by:

Last posted ▾

SF **SI FOU** [27 days ago](#) [Report](#)

Your estimated time is not similar to real time !! what is this error

♡ 0 [Reply](#)

AP **Aleksandr Pervuhin** [20 days ago](#) [Report](#)

The test adds up all the timings from your output and compares with actual time it takes your program to complete. Probably you should try to move variables that check current time so that they cover the program from the very beginning to the very end.

♡ 0 [Reply](#)

MM **Maxim Moliavin** [about 2 months ago](#) [Report](#)

Interesting and rather difficult task but.. Does it actually check bubble sort and jump search? Obviously not.

♡ 0 [Reply](#)