Java 3 Activity 3 Task 3

30003389 – Kyer Potts

Table of Contents

[Table of Contents i](#_Toc25528911)

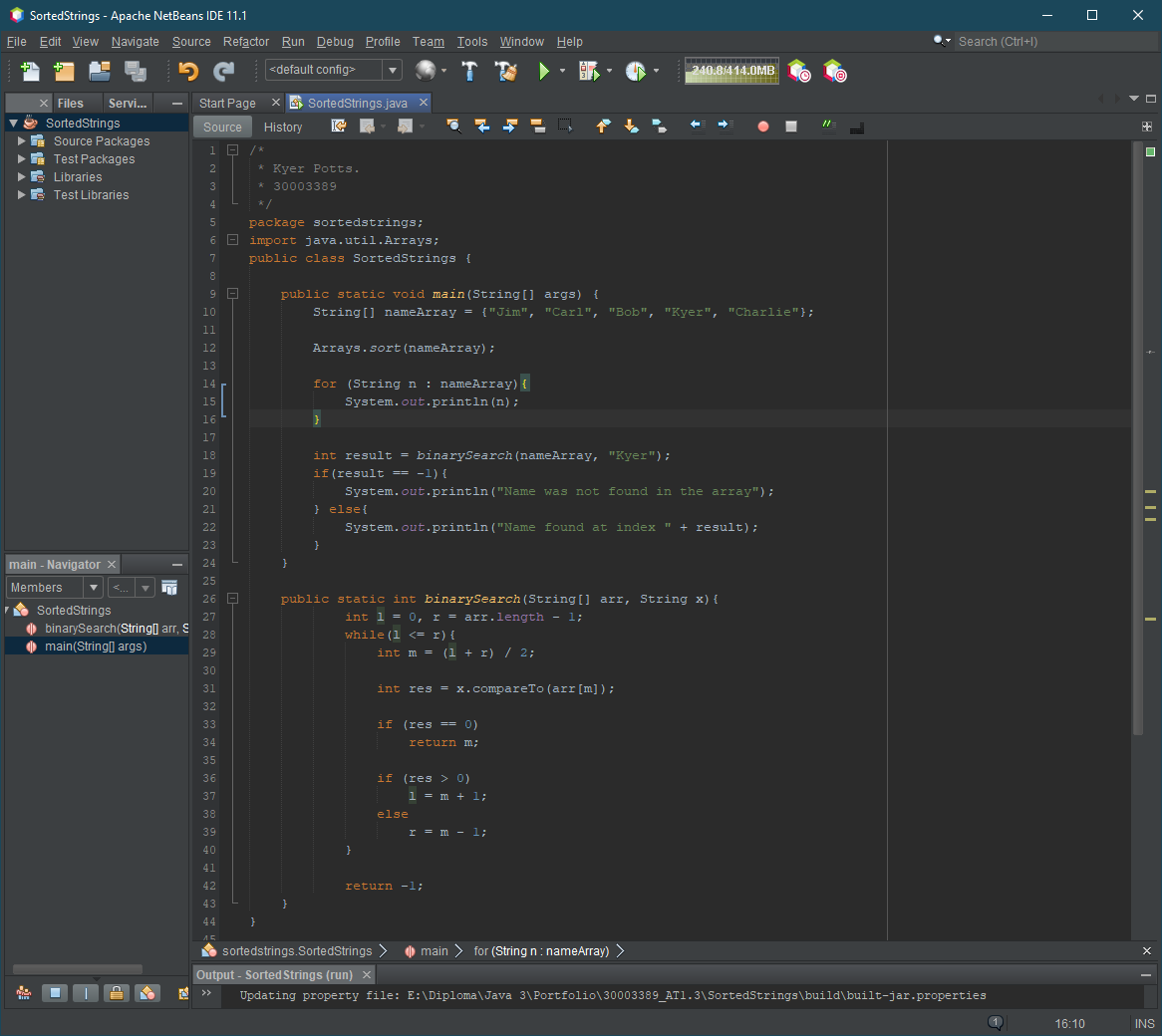
[Sorted String Array 2](#_Toc25528912)

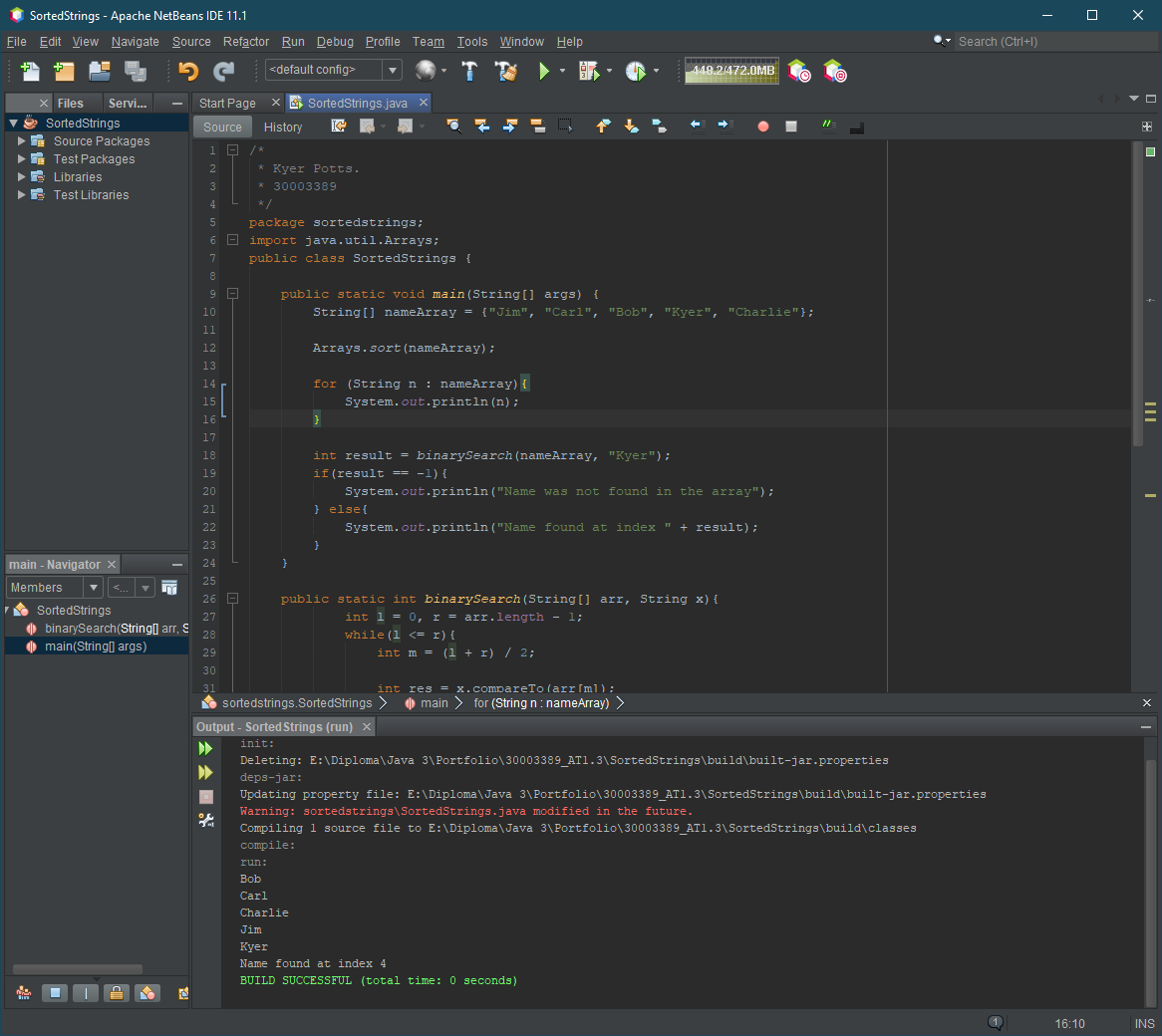
[Complete the following table using Big-O notation. Under the notes section describe which would on average perform the best? 3](#_Toc25528913)

[Describe the difference between a linear and binary search. 3](#_Toc25528914)

[Explain how sorting order is determined if the data contains more than one value. 3](#_Toc25528915)

# Sorted String Array





# Complete the following table using Big-O notation. Under the notes section describe which would on average perform the best?

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Algorithm | Worst | Average | Best | Notes |
| Selection | O(n^2) | Θ(n^2) | Ω(n^2) |  |
| Bubble | O(n^2) | Θ(n^2) | Ω(n) |  |
| Merge | O(n log(n)) | Θ(n log(n)) | Ω(n log(n)) | Mergesort performs the best of these three algorithms due to the time (O) measured against log(n) is much faster than quadratic times achieved by Selection and Bubblesort |

# Describe the difference between a linear and binary search.

A linear search will begin at the start of the data structure and compare the search variable against every element within the structure in order to find a result. Linear searches are most appropriate for unsorted data structures as it allows the search to iterate over all elements instead of separating them into smaller parts.

A binary search will divide the structure in half for each successive search iteration if the search term is not found. A binary search must be performed on a sorted structure, as unsorted structures will have randomly distributed values and elements, and the act of division within this search algorithm may annihilate the search match within the structure.

# Explain how sorting order is determined if the data contains more than one value.

**For example, the student class contains id(int), name(String) and score(double). There are 10 student objects. How do you make the program sort them by the score values in ascending order?**

In order to sort by a specified variable within the element, the comparable interface should be implemented. This allows the element to override the compareTo method and select the variables to be compared for the purposes of the sorting algorithm. The following code snippet contains an example of this in action:

