A03 Autocomplete - Part 2

MISSING 50 Possible Points

| 10/21/2022

Attempt 1

IN PROGRESS
Next Up: Submit Assignment

Add Comment

Unlimited Attempts Allowed

Details

Sorting

Assignment: Autocomplete - Part 2

Learning Objectives

- Determine the performance characteristics of a software application.
- Implement generic methods
- Implement comparators

Overview

In this second part of the assignment, you will complete the program to implement autocomplete for a given set of N strings and positive weights.

Previous (https://slcc.instructure.com/courses/817632/modules/items/18752991)

Submit Assignment

Next (https://slcc.instructure.com/courses/817632/modules/items/18752995)

Continue the project you started in part 1 of this assignment. Ensure that the classes BinarySearchDeluxe and Term work as expected, and implement the class **Autocomplete**.

Assignment Instructions:

The assignment instructions are based on an assignment of Princeton's algorithm course.
http://www.cs.princeton.edu/courses/archive/fall14/cos226/assignments/autocomplete.html (http://www.cs.princeton.edu/courses/archive/fall14/cos226/assignments/autocomplete.html)
They come with additional resources listed below.

You can find JUnit tests for class Autocomplete on CodePost. Again, you can submit as often as you like but only up until the deadline. Before you submit via Canvas, create a screenshot as described before. Ensure that it shows your bruinmail, the class that is tested, and the number of passing/failing tests

Please note, that the JUnit tests on CodePost are only a subset of the tests that are used for grading. They are intended to help you with your testing but they don't cover all cases that need to be tested. It is the responsibility of you and your partner to complete the testing and to ensure that all requirements are met.

Additional Resources:

Whenever there is a difference or inconsistency between the assignment instructions and one of the additional resources, the instructions need to be followed.

Checklist:
Here is a checklist that includes frequently asked questions, input files for testing, and possible progress steps.
http://www.cs.princeton.edu/courses/archive/fall14/cos226/checklist/autocomplete.html (http://www.cs.princeton.edu/courses/archive/fall14/cos226/checklist/autocomplete.html)

Video:
https://www.youtube.com/watch?v=x27BMJ9kGRk&feature=youtu.be (https://www.youtube.com/watch?v=x27BMJ9kGRk&feature=youtu.be)

Previous (https://slcc.instructure.com/courses/817632/modules/items/18752991)

Submit Assignment

Next (https://slcc.instructure.com/courses/817632/modules/items/18752995)

Submission

One team member embeds the screenshot from CodePost and attaches the required java files(no zip, no JAR). **(https://community.canvaslms.com/t5/Student-Guide/How-do-I-embed-images-from-Canvas-into-the-Rich-Content-Editor/ta-p/356)**

Both team members submit the name of the partner and the discussed pebble distribution. If the pebble distribution is not 50/50, include a description that explains the difference.

View Rubric

A03 - Part 2		
Criteria	**Ratings**	**Pts**
JUnit tests for class Autocomplete view longer description		/ 30 pts
Functional and Performance Requirements view longer description		/ 15 pts
Style | Best Practices view longer description		/ 5 pts
		Total Points: 0

Keep in mind, this submission will count for everyone in your Project Groups group.

C

Previous (https://slcc.instructure.com/courses/817632/modules/items/18752991)

Submit Assignment

Next (https://slcc.instructure.com/courses/817632/modules/items/18752995)

Previous (https://slcc.instructure.com/courses/817632/modules/items/18752991)

Submit Assignment

Next (https://slcc.instructure.com/courses/817632/modules/items/18752995)