CMPE 343
Spring 2022
Programming Homework 5

This assignment is due by 23:55 on Wednesday 25, May 2022.

You are welcome to ask your HW related questions. You should use only one of these options:

1. Moodle Homework **Question Forum**: HW Question-and-Answer (Q&A) Forum on Moodle is always available. Use the "Forum" link at the course Moodle page.
2. Homework **RECITATION HOURS**: There will be two Q&A RECITATION HOURs on the following days:

   - CMPE343-HW3 - OfficeHour1: **May 16, 05:00-06:00 PM,** Zoom ID: https://tedu.zoom.us/j/92063819619?pwd=ODBZa1pLRldtd0tPck9DbEpFbjFrUT09

   - CMPE343-HW3 - OfficeHour2: **May 20, 04:00-05:00 PM,** Zoom ID: https://tedu.zoom.us/j/98743627491?pwd=dzI4MUZoUHBhVk50RTd5d3Z2ZUNJdz09

   - CMPE343-HW3 - OfficeHour2: **May 23, 04:00-05:00 PM,** Zoom ID: https://tedu.zoom.us/j/97374143756?pwd=bGRIN2UzR0xxVm9IZHk1NUV3aDUvUT09

Note: Please make sure that you have read the HW document well before participating. However, no HW related questions will be accepted except from the above options.

PROGRAMMING TASK

In this homework, you will compare the four alternatives for implementing Symbol Tables where the key is a string. Several alternatives exist for implementing Symbol Tables, each with their own advantages.

In the homework, you must implement your own functions to perform the following tasks. You need to do the following operations with four data structures: R-way trie, TST, hash tables and binary search tree. You need to compare the performance of these four data structures and explain the performance differences in your report.

- **void FindWord(String arg):** This function should print all strings that start with the given String.

- **int CountAll():** This function should print all words in the txt file and print the count of them.

Here is an example input1.txt:

```
lorem ipsum dolor sit amet consectetur adipiscing
elit
cras est eros pharetra vel vestibulum ut auctor in
odio
maecenas porta pellentesque leo eget vestibulum
in vulputate volutpat nunc
aliquam in gravida tortor eget suscipit erat
maecenas id tristique velit non dictum dui
fusce orci magna pellentesque hendrerit dui eget
elementum dignissim nisl
ut accumsan id erat id efficitur
nullam vitae eros feugiat feugiat augue sed
condimentum dui
aliquam porttitor ipsum eget placerat sollicitudin
aliquam erat volutpat Fusce vitae magna augue
in auctor sit amet orci tempus vehicula
nullam eu imperdiet justo
nulla dui dui accumsan eu suscipit a placerat
bibendum felis
maecenas vitae quam vitae nisl feugiat lacinia
vivamus tincidunt faucibus pretium
quisque vel quam est
vestibulum tempus aliquet diam ut egestas arcu
vestibulum luctus
cras non ornare elit quis tristique nunc
nulla posuere sapien id lectus venenatis vitae mattis
tortor maximus
maecenas molestie placerat tempor
donec luctus facilisis maximus
vivamus quis felis lobortis luctus urna ac
consectetur elit
vivamus scelerisque dui eget massa tincidunt sed
consequat mauris vulputate
in ac pellentesque ante
proin vel mattis ipsum
donec rhoncus ex leo eu porta mauris blandit id
aliquam blandit hendrerit nulla eu bibendum lorem
vulputate nec
praesent interdum urna sit amet leo convallis varius
```

Here is example output:

```
input1.txt
FindWord con
consectetur condimentum consequat convallis
```

Here is an example input2.txt:

```
lorem ipsum dolor sit amet consectetur adipiscing
elit praesent suscipit convallis nibh maximus luctus
quis lacus nunc leo vel faucibus nunc nec augue
elementum lorem convallis non nunc enim cursus
```

Here is an example output:

```
input2.txt
CountAll
lorem:2
ipsum:1
dolor:1
sit:1
amet:1
consectetur:1
adipiscing:1
elit:1
praesent:1
suscipit:1
convallis:2
nibh:1
maximus:1
luctus:1
quis:1
lacus:1
nunc:3
leo:1
vel:1
faucibus:1
nec:1
augue:1
elementum:1
non:1
enim:1
cursus:1
```

You need to perform these 3 operations by using 4 data structures: R-way trie, TST, BST, and hash tables. You can use the code in the textbook web site as the basis. You must not use any other implementations than those available at our course textbook web site.

Your report should explain the performance difference of these data structures according to time complexity of the operations in your report. **You should address both time and space complexity of these solutions.**

If you think a data structure cannot be used for a particular task, you may skip it and explain in your report.

WHAT TO HAND IN

A zip file for both parts containing:

- The Java sources for your program.
- The Java sources should be WELL DOCUMENTED as comments, as part of your grade will be based on the level of your comments.
- You can test your Java source files on available Moodle VPL environment to ensure your solution's correctness before submitting. VPL simply tests your program's output by checking against given sample input.
- A **maximum-3 pages** PDF report document that explains your own answers for programming task in a clearly readable PA report format (refer to **PA REPORT FORMAT** section).

PA REPORT FORMAT

A programming assignment report is a self-description of a programming assignment and your solution. The report must not be hand-written. You may use a word processor or the on-line editor of your choice and prepare as a PDF document. The report must be grammatically correct and use complete English sentences. Each report should include the following sections, in the order given:

Information (%2.5): This section includes your ID, name, section, assignment number information properly.

Problem Statement and Code Design (%15): Include a brief summary of the problem and/or your sub-tasks to be completed in this assignment. You should show your modular design rationale by creating a structure chart that indicates your top-down, stepwise refinement of the problem solution. You may create the structure chart using available graphical tools like MS PowerPoint, SmartDraw etc.

Implementation, Functionality(%20): Since you have modular source code, you should describe each sub-module (program) in this section. Each sub-module should include names and types of any input/output parameters as well as the pseudocode algorithm that used for completing its task. By this way, you give meaning to each chart boxes from the previous section. Also, you should add your performance comparison, part II, here.

Testing (%7.5): You should provide a tester class that is able to identify key test points of your program. This class should be able to generate additional (apart from the given sample input/output) test data for the purpose of being clear on what aspects of the solution are being tested with each set. This section should also include a description of any program *bugs* that is, tests which has incorrect results. You should write these to describe your tests, summarize your results, and argue that they cover all types of program behavior.

Final Assessments (%5): In this final section, you should briefly answer the following questions:

- What were the trouble points in completing this assignment?
- Which parts were the most challenging for you?
- What did you like about the assignment? What did you learn from it?

GRADING:

- Codes (%50: %25 for Q1 and %25 for Q2)
  - Available test cases evaluation on VPL: %15
  - Hidden test cases evaluation: %15
  - Approach to the problem: %20
- Report (%50: %25 for Q1 and %25 for Q2)
  - Information: %2.5
  - Problem Statement and Code design: %15
  - Implementation, Functionality: %20
  - Testing: %7.5
  - Final Assessments: %5

IMPORTANT

IMPORTANT NOTES: Do not start your homework before reading these notes!!!

1. This assignment is due by **23:55 on Wednesday, May 25th**.

2. You should upload your homework to Moodle before the deadline. No hardcopy submission is needed. You should upload files and any additional files if you wrote additional classes in your solution as a single archive file (e.g., zip, rar).

3. The standard rules about late homework submissions apply (20 points will be deducted for each late day). Please see the course syllabus for further discussion of the late homework policy as well as academic integrity.

4. You ARE NOT ALLOWED to modify the given method names. However, if necessary, you may define additional data members and member functions.

5. Your classes' name MUST BE as shown in the homework description.

6. The submissions that do not obey these rules will not be graded.

7. To increase the efficiency of the grading process as well as the readability of your code, you have to follow the following instructions about the format and general layout of your program.

8. Do not forget to write down your id, name, section, assignment number or any other information relevant to your program in the beginning of your Java files. Example:

```
//-----------------------------------------------------
// Title: Scheduler tester class
// Author: Name/Surname
// ID: 2100000000
// Section: 1
// Assignment: 1
// Description: This class tests the …
//-----------------------------------------------------
```

9. Since your codes will be checked without your observation, you should report everything about your implementation. Add detailed comments to your classes, functions, declarations etc. Make sure that you explain each function in the beginning of your function structure. Example:

```
void setVariable(char varName, int varValue)
//-----------------------------------------------------
// Summary: Assigns a value to the variable whose
// name is given.
// Precondition: varName is a char and varValue is an
// integer
// Postcondition: The value of the variable is set.
//-----------------------------------------------------
{
      // Body of the function
}
```

10. Indentation, indentation, indentation...

11. This homework will be graded by your TA Deniz Merve Gündüz. Thus, you may ask them your homework related questions through HW forum on Moodle course page. You are also welcome to ask your course instructors Tolga Çapın for help.