CS300 – Spring 2021-2022 - Sabancı University

Homework #2 – Notebook

Due: 13 April, Wednesday, 23:45

Brief Description

In this homework, you will implement two "Notebooks" by using the AVL Search Tree and the Binary Search Tree (BST). Additionally, you will display the speed of some of the operations on these tree data structures that are used for the implementation. Each "Notebook" will have a nested tree structure, which will be explained in more detail in the rest of the documentation.

Although each Notebook tree will have a separate implementation, they will store the same data and will perform the same operations. Basically, the only difference between the Notebooks will be the data structures that are going to be used for the implementation. Each Notebook will contain multiple *sections* and each *section* will have multiple *items* which will store the information inside.

A notebook will have a nested structure as follows:

- **Notebook**
  May have one or many **section**/s.
  - **Section**
    Will have a **title**.
    May have one or many **item**/s.
    - **Item**
      Will have a **title**.
      Will have information.

Example Notebook:

```
Courseworks
      CS300-the homework is due Sunday
      CS408-don't forget the quiz
      CS303-16:40 office hour
Movies
       The Holy Mountain-7.9
       I Lost My Body-7.6
Phonebook
      Gulsen-09990000099
      Alp-09998765432
      Furkan-09876543210
      Serra-09898989898
```

As seen in the example above, a notebook is a tree of sections. Each section has got a title string and a tree of items. Each item has a title string and an information string.

There cannot be two sections with the same title inside the same notebook. Also, there cannot be two items with the same title inside the same section.

BST and AVL Tree

You are going to implement a BST class and an AVL Tree class. Which will get used separately to implement two different notebooks that will perform the same operations over the same data. You **CANNOT USE VECTOR OR ARRAY** to implement the notebooks. Also, you **cannot use a BST inside an AVL Tree and vice versa**. In other words, there will be two notebooks, one of them will be implemented by using nested AVL Trees and the other will be implemented by using the nested BSTs only.

Both the BST and the AVL Tree classes should be reusable for different types, classes, etc. Thus, both the BST and AVL Tree classes **MUST** be **template-based** classes.

Use the "**title"** properties of the section and the item to compare the nodes inside the trees.

Program Flow

- **Initializing the notebooks**

  At the very beginning of the program, you should create two notebooks (one by using the BST, other by using the AVL Tree).

  Then, your program will read a text file that contains the section and item data of the notebook. Your program will insert the data into the notebooks accordingly.

  Your program will display the elapsed insertion time of each section for both of the notebooks. Then the program will direct to the main menu.

- **Main menu**

  On the main menu, there will be 6 options. The user will enter an input between [1-6] then the program will perform the commands on both notebooks accordingly. These options are:

  1. Display the sections (AVL)

     Your program should perform an inorder traversal over the AVL notebook and print the section titles in ascending order (alphabetically).
  2. Display the sections (BST)

     Your program should perform an inorder traversal over the BST notebook and print the section titles in ascending order (alphabetically).
  3. Select a section

     The program will get a section title name from the user then it will search for the section with the same title, both in the AVL and BST notebooks.
     If there exists no such section your program will display an error. Else it will direct to the **section menu** to perform operations on the selected section.

4. Add new section
   Get a title input from the user. If a section with the given title already exists in the notebook, display an error message. Else, create a new section with that title and insert it into the notebooks.
5. Delete a section
   Get a title input from the user. If a section with the given title does not exist in the notebook, display an error message. Else, remove the section with that title from the notebooks.
6. Exit
   Terminate the program.

❑ **Section menu**

At option 3 of the main page, if the title of any section matches with the input from the user, the program should land on the section menu page to perform operations on the selected section.

On the main menu, there will be 7 options. The user will enter an input between [1-7] then the program will perform the commands on both notebooks accordingly. Options are:

1. Display the items (AVL)
   For the selected section of the AVL notebook:
   Your program should perform an inorder traversal over the items tree of the section and print the item titles in ascending order (alphabetically).
2. Display the items (BST)
   For the selected section of the BST notebook:
   Your program should perform an inorder traversal over the items tree of the section and print the item titles in ascending order (alphabetically).
3. Display the information of an item
   Get an item title input from the user. If an item with the given title does not exist in the notebook, display an error message. Else, display the information of the matching item.

   Note: Although, you should make a query on both the AVL notebook and the BST notebook you can display the matching item information once. Since, if your program works correctly, matching item information of the AVL notebook and the BST notebook will be the same.

   In both cases, display the elapsed query/insertion time for the AVL notebook and the BST notebook.
   Display the elapsed query time.

4. Add a new item
   Get an item title input from the user. If an item with the given title already exists in the section, display an error message. Else, create a new item with that title and insert it into the selected section.
   In both cases, display the elapsed query/insertion time.
5. Update the information of an item:
   Get an item title input from the user. If an item with the given title does not exist in the section, display an error message. Else, get the new information from the user and update the information of the item accordingly.
   Display the elapsed query/update time.
6. Delete an item
   Get an item title input from the user. If an item with the given title does not exist in the section, display an error message. Else, remove the item with that title from the section. Display the elapsed deletion time.
7. Return to main menu
   Go back to the main menu.

Text File

As mentioned before we have provided initial data to fill the notebooks at the beginning of the program. You should read the data from a text file (.txt extension) and insert it into the notebooks accordingly.

The text file will contain the following 3 sections:

- Courseworks: contains 5 non-sorted items.
- Movies: contains +90000 non-sorted items.
- Phonebook: contains +6000 **sorted** items.

❑ **Data format**

If the first character of a line is not a '-' (dash), it is the title of a new section. Until the occurrence of a new section, as long as a new line starts with a dash it denotes the item of the given section. Each item line contains exactly two dashes. The first one denotes that it is an item line and the second one separates the item title from the item info.

So, a text file with notebook data inside it will look like the following:

```
Section title 1
-item title 1-item info 1
-item title 2-item info 2
-item title 3-item info 3
Section title 2
-item title 1-item info 1
-item title 2-item info 2
```

Example text file:

```
Courseworks
-CS300-the homework is due Sunday
-CS408-don't forget the quiz
-CS303-16:40 office hour
Movies
-The Holy Mountain (1973)-7.9
-Climax-7
-Midsommar-7.6
```

Elapsed Time

You should display the elapsed time of the notebook operations. For that purpose, you can use the C++ built-in library **chrono**. You can include the library by using `#include <chrono>`.

You are recommended to use **`chrono::high_resolution_clock`** to obtain accurate results. Documentation: https://en.cppreference.com/w/cpp/chrono/high_resolution_clock

Here is an example to get the time duration of a foo() function in microseconds:

```
auto start = std::chrono::high_resolution_clock::now(); // start time
foo();
auto end = std::chrono::high_resolution_clock::now(); // end time
long long timer = (end-start).count() / 1000.0; // get the elapsed time in micro seconds
std::cout << "Elapsed time of the foo: " << timer << " µs" << std::endl; // display the elapsed time
```

Between the start and the end clocks, you should only include the member functions of the trees (insert, delete, update, etc.) and exclude the rest of the code (getline, loops, arithmetic operations, etc.).

Sample Run

```
Welcome to the Notebook!

Section "Courseworks" has been inserted into the AVL notebook.
[AVL] Elapsed time: 15 microseconds
Section "Courseworks" has been inserted into the BST notebook.
[BST] Elapsed time: 7 microseconds

Section "Phonebook" has been inserted into the AVL notebook.
[AVL] Elapsed time: 18236 microseconds
Section "Phonebook" has been inserted into the BST notebook.
[BST] Elapsed time: 3864642 microseconds

Section "Movies" has been inserted into the AVL notebook.
[AVL] Elapsed time: 300933 microseconds
Section "Movies" has been inserted into the BST notebook.
[BST] Elapsed time: 353231 microseconds

MENU
Please enter an input between [1 - 6]:
1- Display the sections [AVL]
2- Display the sections [BST]
3- Select a section
4- Add new section
5- Delete a section
6- Exit
Input: 1

*****
Courseworks
Movies
Phonebook
*****

Input: 2

*****
Courseworks
Movies
Phonebook
*****

Input: 3
Enter the title of the section: Cour
Invalid title!

Input: 3
Enter the title of the section: Courseworks
```

```
Selected section -> Courseworks
Please enter an input between [1 - 7]:
1- Display the items [AVL]
2- Display the items [BST]
3- Display the information of a item
4- Add new item
5- Update the information of a item
6- Delete an item
7- Return to main menu
Input: 1

*****
CS300
CS303
CS408
ENS492
SPS303
*****

Input: 2

*****
CS300
CS303
CS408
ENS492
SPS303
*****

Input: 3
Enter the title of the item: CS100
[AVL] Elapsed time: 8 microseconds
[BST] Elapsed time: 2 microseconds
Invalid title.

Input: 3
Enter the title of the item: CS300
[AVL] Elapsed time: 6 microseconds
[BST] Elapsed time: 0 microseconds
the homework is due Sunday

Input: 4
Enter a title for the item: CS303
Item "CS303" already exists in the "Courseworks".

Input: 4
Enter a title for the item: CS600
Enter a description for the item: huh?
[AVL] Elapsed time: 14 microseconds
[BST] Elapsed time: 3 microseconds
The new item "CS600" has been inserted.
```

Input: **5**
Enter the title of the item: **CS999**
[AVL] Elapsed time: 5 microseconds
[BST] Elapsed time: 2 microseconds
Item "CS999" does not exist in the "Courseworks".

Input: **5**
Enter the title of the item: **CS300**
[AVL] Elapsed time: 6 microseconds
[BST] Elapsed time: 0 microseconds
Enter the new information: **great**
The content CS300 has been updated.

Input: **3**
Enter the title of the item: **CS300**
[AVL] Elapsed time: 7 microseconds
[BST] Elapsed time: 1 microseconds
great

Input: **6**
Enter the title of the item: **CS123**
Item "CS123" does not exist in the "Courseworks".

Input: **6**
Enter the title of the item: **CS300**
[AVL] Elapsed time: 2 microseconds
[BST] Elapsed time: 2 microseconds
The item "CS300" has been deleted.

Input: **1**

*****
CS303
CS408
CS600
ENS492
SPS303
*****

Input: **2**

*****
CS303
CS408
CS600
ENS492
SPS303
*****

Input: **7**
MENU
Please enter an input between [1 - 6]:

```
1- Display the sections [AVL]
2- Display the sections [BST]
3- Select a section
4- Add new section
5- Delete a section
6- Exit
Input: 4
Enter a title for the section: Courseworks
Section "Courseworks" already exists.

Input: 4
Enter a title for the section: Colors
The new section "Colors" has been inserted.

Input: 3
Enter the title of the section: Colors

Selected section -> Colors
Please enter an input between [1 - 7]:
1- Display the items [AVL]
2- Display the items [BST]
3- Display the information of a item
4- Add new item
5- Update the information of a item
6- Delete an item
7- Return to main menu
Input: 1

*****
*****

Input: 2

*****
*****

Input: 4
Enter a title for the item: Red
Enter a description for the item: Fav
[AVL] Elapsed time: 4 microseconds
[BST] Elapsed time: 1 microseconds
The new item "Red" has been inserted.

Input: 1

*****
Red
*****
```

Input: **2**

*****
Red
*****

Input: **7**

MENU
Please enter an input between [1 - 6]:
1- Display the sections [AVL]
2- Display the sections [BST]
3- Select a section
4- Add new section
5- Delete a section
6- Exit
Input: **1**

*****
Colors
Courseworks
Movies
Phonebook
*****

Input: **2**

*****
Colors
Courseworks
Movies
Phonebook
*****

Input: **5**
Enter the title of the section: **Course**
Section "Course" does not exist.

Input: **5**
Enter the title of the section: **Courseworks**
The section has been deleted.

Input: **1**

*****
Colors
Movies
Phonebook
*****

Input: **2**

*****
Colors
Movies
Phonebook
*****

Input: **3**
Enter the title of the section: **Phonebook**

Selected section -> Phonebook
Please enter an input between [1 - 7]:
1- Display the items [AVL]
2- Display the items [BST]
3- Display the information of a item
4- Add new item
5- Update the information of a item
6- Delete an item
7- Return to main menu
Input: **3**
Enter the title of the item: **Aada**
[AVL] Elapsed time: 2465 microseconds
[BST] Elapsed time: 0 microseconds
0016839979200

Input: **3**
Enter the title of the item: **Zyrie**
[AVL] Elapsed time: 2302 microseconds
[BST] Elapsed time: 2276 microseconds
0017707501460

Input: **5**
Enter the title of the item: **Zyrie**
[AVL] Elapsed time: 2410 microseconds
[BST] Elapsed time: 2327 microseconds
Enter the new information: ***000000***
The content Zyrie has been updated.

Input: **6**
Enter the title of the item: **Kadeem**
[AVL] Elapsed time: 6 microseconds
[BST] Elapsed time: 878 microseconds
The item "Kadeem" has been deleted.

Input: **4**
Enter a title for the item: **Zxxy**
Enter a description for the item: **???**
[AVL] Elapsed time: 12 microseconds
[BST] Elapsed time: 2143 microseconds
The new item "Zxxy" has been inserted.

Input: **7**

MENU
Please enter an input between [1 - 6]:
1- Display the sections [AVL]
2- Display the sections [BST]
3- Select a section
4- Add new section
5- Delete a section
6- Exit
Input: **3**
Enter the title of the section: **Movies**

Selected section -> Movies
Please enter an input between [1 - 7]:
1- Display the items [AVL]
2- Display the items [BST]
3- Display the information of a item
4- Add new item
5- Update the information of a item
6- Delete an item
7- Return to main menu
Input: **3**
Enter the title of the item: **Zingaresca**
[AVL] Elapsed time: 31739 microseconds
[BST] Elapsed time: 2 microseconds
6.6

Input: **5**
Enter the title of the item: **Aurora**
[AVL] Elapsed time: 30734 microseconds
[BST] Elapsed time: 2 microseconds
Enter the new information: **cool**
The content Aurora has been updated.

Input: **6**
Enter the title of the item: **Redskin**
[AVL] Elapsed time: 8 microseconds
[BST] Elapsed time: 1 microseconds
The item "Redskin" has been deleted.

Input: **7**

MENU
Please enter an input between [1 - 6]:
1- Display the sections [AVL]
2- Display the sections [BST]
3- Select a section
4- Add new section
5- Delete a section
6- Exit
Input: **6**

General Rules and Guidelines about Homeworks

The following rules and guidelines will apply to all homework unless otherwise noted.

How to get help?

You may ask questions to TAs (Teaching Assistants) of CS300. Office hours of TAs can be found here. Recitations will partially be dedicated to clarifying the issues related to homework, so it is to your benefit to attend recitations.

What and Where to Submit

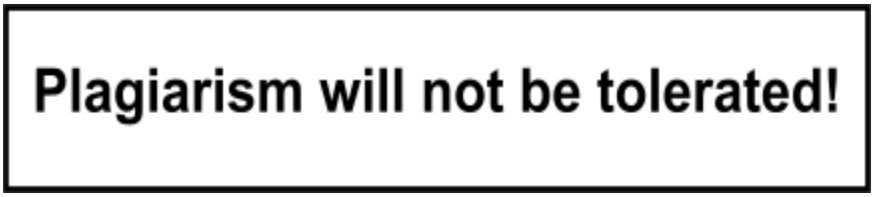
Please see the detailed instructions below/on the next page. The submission steps will get natural/easy for later homework.

Grading and Objections

Careful about the semi-automatic grading: Your programs will be graded using a semi-automated system. Therefore, you should follow the guidelines about input and output order; moreover, you should also use the exact same prompts as given in the Sample Runs. Otherwise, the semi-automated grading process will fail for your homework, and you may get a zero, or in the best scenario, you will lose points.

Grading:

- Late penalty is 10% off the full grade and only one late day is allowed.
- **Having a correct program is necessary, but not sufficient to get the full grade. Comments, indentation, meaningful and understandable identifier names, informative introduction and prompts, and especially proper use of required functions, unnecessarily long programs (which is bad) and unnecessary code duplications will also affect your grade.**
- Please submit your own work only (even if it is not working). It is really easy to find out "similar" programs!
- For detailed rules and course policy on plagiarism, please check out http://myweb.sabanciuniv.edu/gulsend/courses/cs201/plagiarism/



Grade announcements: Grades will be posted in SUCourse, and you will get an Announcement at the same time. You will find the grading policy and test cases in that announcement.

Grade objections: Since we will grade your homework with a demo session, there will be very likely no further objection to your grade once determined during the demo.

What and where to submit (IMPORTANT)

Submission guidelines are below. Most parts of the grading process are automatic. Students are expected to strictly follow these guidelines to have a smooth grading process. If you do not follow these guidelines, depending on the severity of the problem created during the grading process, 5 or more penalty points are to be deducted from the grade.

Add your name to the program: It is a good practice to write your name and last name somewhere in the beginning program (as a comment line of course).

Name your submission file:

- Use only English alphabet letters, digits or underscore in the file names. Do not use a blank, Turkish characters or any other special symbols or characters.
- Name your cpp file that contains your program as follows.

  "SUCourseUserName_yourLastname_yourName_HWnumber.cpp"
- Your SUCourse user name is your SUNet username which is used for checking sabanciuniv e-mails. Do NOT use any spaces, non-ASCII and Turkish characters in the file name. For example, if your SUCourse user name is cago, name is Çağlayan, and last name is Özbugsızkodyazaroğlu, then the file name must be:

  cago_ozbugsizkodyazaroglu_caglayan_hw2.cpp
- Do not add any other character or phrase to the file name.
- Make sure that this file is the latest version of your homework program.
- You need to submit ALL .cpp and .h files including the data structure files in addition to your main.cpp in your project.

Submission:

Submit via SUCourse+ ONLY! You will receive no credits if you submit by other means (e-mail, paper, etc.).

Successful submission is one of the requirements of the homework. If for some reason, you cannot successfully submit your homework and we cannot grade it, your grade will be 0.

Good Luck!
Gülşen Demiröz