


# Assignments

## Assignment - In progress

Add attachment(s), then choose the appropriate button at the bottom.

In progress  
Submitted  
Returned  
Title  
A4 Binary Search Trees   
Due  
19 Apr 2023 11:55 PM  
Number of resubmissions allowed  
Unlimited  
Accept Resubmission Until  
24 Apr 2023 11:55 PM  
Status  
Not Started  
Grade Scale  
Points (max 100.00)  
Modified by instructor  
15 Apr 2023 11:07 AM

### Instructions

## CSC2001F 2023 Assignment 1

### Instructions

Tiktok is a very popular social media application centred on the creation and sharing of video clips. In March 2023, the application came under scrutiny because of political reasons and the US government threatened to ban its use. If this happens, it is very likely that programmers will scramble to create similar applications to fill the void.

The goal of this assignment is to produce a simple proof-of-concept replica of some functionality of a social media system like Tiktok, using Binary Search Trees as the main underlying data structure. We assume for this assignment that data is stored in memory while the application performs multiple functions on-demand until the user exits.

### Application

In your application, every user has a record with 3 data fields: account, profile description and an associated list of posts (most recent first). A post has at least a title, name of video clip file and number of likes.

Your application must include at least the functionality to do the following:

- Find the profile description for a given account
- List all accounts (in any order)
- Create an account
- Delete an account
- Display all posts for a single account (with newest posts first)
- Add a new post for an account
- Load a file of actions from disk and process this in sequence. These are only Create and Add actions.

You MUST use a Binary Search Tree (BST; balanced or not balanced) to implement the data structure for storing accounts. Your BST implementation can be created from scratch or re-used from anywhere.

You may use any user interface for the application - at least a text menu is required but the interface can be graphical or GUI-based. For example:

Choose an action from the menu:

1. Find the profile description for a given account

- 2. List all accounts
  - 3. Create an account
  - 4. Delete an account
  - 5. Display all posts for a single account
  - 6. Add a new post for an account
  - 7. Load a file of actions from disk and process this
  - 8. Quit
- Enter your choice: 1

Enter the account name: shouzichew  
The profile description is: Shou Zi Chew is the CEO of TIKtok.

- Choose an action from the menu:
- 1. Find the profile description for a given account
  - 2. List all accounts
  - 3. Create an account
  - 4. Delete an account
  - 5. Display all posts for a single account
  - 6. Add a new post for an account
  - 7. Load a file of actions from disk and process this
  - 8. Quit
- Enter your choice: 5

Enter the account name: shouzichew  
Title: More cats playing with wool  
Video: video2.mpg  
Number of likes: 75  
Title: Cats playing with wool  
Video: video1.mpg  
Number of likes: 34

- Choose an action from the menu:
- 1. Find the profile description for a given account
  - 2. List all accounts
  - 3. Create an account
  - 4. Delete an account
  - 5. Display all posts for a single account
  - 6. Add a new post for an account
  - 7. Load a file of actions from disk and process this
  - 8. Quit
- Enter your choice: 8

Bye!

You should be able to invoke your application using a command such as:

```
java -cp bin TokTik
```

Test your program thoroughly and ensure that it functions correctly for different inputs. Also, output error messages where appropriate (e.g., when a Find action fails to find a matching account name).

Hint: Do this assignment incrementally. First create a data structure with only account names and descriptions and get the related functions working one at a time. Then add in the functions related to posts. Finally, write the code to load in the external file.

### Dataset for file loading

The function to load actions from disk must process a file with lines such as the following:

Create hussein The lecturer dude.  
Add hussein video34.mpg 34 Yet another video of cats  
Add hussein video45.mpg 53 Omg more video of cats

This format assumes that accounts and video filenames are single words with no spaces. The integer after the filename is the number of likes. With both actions, the rest of each line is the description or title.

The attached file (dataset.txt) is a sample file that you can load. The associated script is also included to generate this on an Ubuntu system.

You can experiment with larger and smaller data files - your program should work with fairly large datasets because of the efficiency of the BST.

Study the data carefully. The data loading should be relatively straight-forward and you MUST write your own code to read in the text file.

Report

Write a report that includes the following:

- What your OO design is: what classes you created, why, and how they interact (at most 1 page).
- What test values you used during testing and what the output was in each case (use output redirection or cut-and-paste or take screenshots) (at most 10 pages).
- A statement of what you included in your application(s) that constitutes creativity - how you went beyond the basic requirements of the assignment (at most 1 page). Examples of creativity include: designing a visual/multimedia interface and implementing a follower/followed feature for multiuser feeds.
- Summary statistics from your use of git to demonstrate usage. Print out the first 10 lines and last 10 lines from "git log" , with line numbers added. You can use a Unix command such as:

```
git log | (ln=0; while read l; do echo $ln\: $l; ln=$((ln+1)); done) | (head -10; echo ...; tail -10)
```

Dev requirements

As a software developer, you are required to make appropriate use of the following tools:

- git, for source code management
- javadoc, for documentation generation
- make, for automation of compilation and documentation generation

Submission requirements

Submit a .tar.gz compressed archive containing:

- Makefile
- src/
  - all source code
- bin/
  - all class files
- doc/
  - javadoc output
- report.pdf

Your report must be in PDF format. Do not submit the git repository.




Marking Guidelines

Your assignment will be marked by tutors, using the following marking guide.

Artefact Aspect	Mark
Report Appropriate design and implementation of OOP and data structures	10
Report Trial test values and outputs (demonstrating all 6 functions)	10
Creativity	10

	Git usage log	5
Code	Works correctly for all 7 functions	50
Dev	Documentation - javadoc	10
	Makefile - make and clean targets (docs is optional)	5

Additional resources for assignment

-  [makedataset.py](#) ( 2 KB; 29-Mar-2023 19:32 )
-  [names.txt](#) ( 1 KB; 29-Mar-2023 19:32 )
-  [dataset.txt](#) ( 547 KB; 29-Mar-2023 19:32 )

Submission

Attachments


No attachments yet

Select a file from computer  No file chosen

Preview

Save Draft

Cancel

 Don't forget to save or proceed!