MIDDLE EAST TECHNICAL UNIVERSITY, NORTHERN CYPRUS CAMPUS

CNG213 C Programming – Programming Assignment 4

Date handed out: **Sunday 27 December 2015**

Date submission due: **Friday 08 January 2016**

“Indexing Tweets!”

This assignment aims to help you practice binary search tree ADT, in particular AVL Tree. You will write a program that creates an index of a given list of tweets.

Requirements

In this assignment, you are given a list of tweets in an external text file called “tweets.txt”. In this text file you can find a number of real tweets extracted from Twitter[1]. Most of the tweets these days include a hashtag[2] which start with the “#” character. For example “#programming” hashtag can be used by somebody who sends a tweet related to programming and then when somebody else also sends a tweet with the same hashtag, these tweets are grouped together.

In this assignment, we want to process the given tweets and find out the hashtags used in these tweets and index these tweets given these hashtags. In order to create such index, we will process these tweets and whenever we find a hashtag (i.e., a text which starts with the “#” character and finishes with a space), we will add that hashtag to the AVL tree and we will also record the tweet number. For example, when we process the first tweet (first line), which has the following tweet:

*“Build a Website by amith7951 https://t.co/eoW5q7V2Fh #**net** #**ajax** #**asp** #**cprogramming** #**sql**”*

we will add “net” “ajax” “asp” “cprogramming” and “sql” to the AVL tree and for each node we will also record the tweet number 1. Then when we process the second tweet:

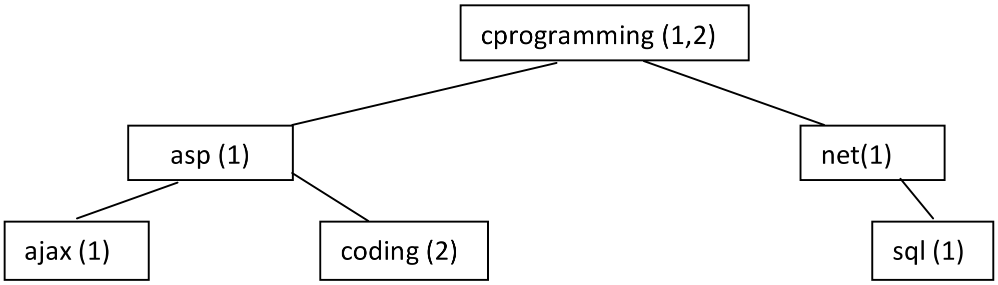
*“Program Entirely with Static Methods. Great post from @mikehadlow https://t.co/qzTxx6c22t #**cprogramming** #**coding**”*

we will need to record “cprogramming” and “coding” but “cprogramming” is already recorded so we will only add tweet number 2 to this node in the tree, and we will insert “coding” to the AVL tree.

In summary, after we process the first two tweets, our AVL tree will look like this.

[1] https://twitter.com/

[2] (On social media sites such as Twitter) a word or phrase preceded by a hash or pound sign (#) and used to identify messages on a specific topic <http://www.oxforddictionaries.com/us/definition/american_english/hashtag>



Once you finish indexing the tweets in the document, then you will display the index of the tweets by using the AVL tree. You will need to display all the hashtags indexed in the alphabetical order. For example, for the given AVL tree above, you will display the following:

ajax: tweet:1

asp: tweet: 1

coding: tweet: 2

cprogramming: tweet: 1, 2

net: tweet: 1

sql: tweet: 1

Programming Requirements:

You will start this programming by taking the file name as a command line input and then you will need to implement at least the following functions:

`read-tweet-data`: This function will mainly process the external file. As an input it will take the file name and it will return an AVL tree.

`insert-hashtag`: This function will take an AVL tree, a hashtag, and the tweet number and it will try to insert it to the AVL tree. If it is already in the tree then it will update the existing node with the tweet number (you cannot again make assumptions about the tweet number here, so make sure that you use a dynamic list – you are encouraged to use a linked list here), if it is not then it will create a new node and add it to the tree.

`display-index`: This function will mainly take an AVL tree and display the index of the tweets in an alphabetical order.

Please note that in this assignment, you can make use of the functions in the string.h library and similar external libraries. Please also note that your solution should be able to take any file in the given format and be able to create the index. You cannot make an assumption about the number of tweets given in this external file.

Submission Requirements:

Create a project under the **CNG213_Assignment_4** folder. Separate the major functionality of your Abstract Data Types into header files and put them under the **CNG213_Assignment_4/Project_Lib** folder. Project submission should be compressed version of **CNG213_Assignment_4** folder. If you do not follow this structure, you will loose %10 from the overall grade.

Programming Style Tips!

Please follow the modular programming approach. In C we use functions referred to modules to perform specific tasks that are determined/guided by the solution. Remember the following tips!

- Modules can be written and tested separately!
- Modules can be reused!
- Large projects can be developed in parallel by using modules!
- Modules can reduce the length of the program!
- Modules can also make your code more readable!

Grading

Your program will be graded as follows:

Grading Point	**Mark (100)**
Processing tweet data txt file (`read-tweet-data()`)	30 pts
Creating a structure for each AVL tree node that will store both hashtag and the list of tweet numbers	10 pts
AVL tree functions including `insert-hashtag` () and all the auxiliary functions such as searching in AVL tree	50 pts
Displaying the index (`display_index()`)	10 pts

NOTE: Remember to have good programming style (Appropriate comments, variable names, formulation of selection statements and loops, reusability, extensibility etc.). Each of the items above will include10% for good programming style. Good programming style also includes modularity approach explained above.