CS 202 Fundamental Structures of Computer Science II
Assignment 4 – AVL, 2-3 and 2-3-4 Trees

Assigned on: 29 November 2016 (Tuesday)
Due Date: 09 December 2016 (Friday) – 23:55

Question-1 (30 Points)

For questions below, make sure to use the operations as described in lecture slides.

a) (*7.5 points*) Insert D, I, J, K, S, T, R, A to an empty 2-3 tree, and then delete S and I in given order. Show the evolution of the 2-3 tree after each insertion and deletion operation.
b) (7.*5 points*) Insert V, I, N, T, C, E, R, F to an empty 2-3-4 tree, and then delete T and N in given order. Show the evolution of the 2-3-4 tree after each insertion and deletion operation.
c) (*5 points*) What is the maximum number of keys that a 2-3 tree of height *h* can hold?
d) (*10 points*) Let *S* and *T* be the 2-3 trees of same height *h*. Suppose every key of *S* is smaller than every key of *T*. Write a pseudocode for a function MERGE(*S*, *T*) that runs in *O(h)* time and returns a 2-3 tree that contains all of the nodes of *S* and *T*.

Question-2 – Help People of Meereen (70 Points)

Suppose that you are a clever scientist and advisor to the ruler, Daenerys Targaryen. After conquering the city of Meereen, Daenerys decides to change the language of the city to Dothraki language. The people of Meereen do not know this language, but they try to learn. However, since this language is all new to these people, they prone to making mistakes while using the vocabulary of the Dothraki language. Daenerys notices this problem and wants you to design a program that will help people on usage of vocabulary. This program will check whether a word is in vocabulary and if it is not, it will give some suggestions based on the searched word.

Since this task will be mainly based on search operations, you think that using AVL tree structure is an appropriate decision. You are provided with the vocabulary of Dothraki language in a text file[1]. You will read each word and then insert to the AVL tree in order given in the file. By using constructed AVL tree, you will be able to design an efficient way to make decision about the existence of the word and give suggested words to the user, if necessary. You will not delete any word from the vocabulary list, therefore there is no need to write a remove function for the AVL tree.

If the word searched by the user is not in the vocabulary, the suggested words should be according to the maximum number of similar letters starting from the first letter of the searched word (i.e. a longest prefix match). In other words, we should look for words that have the same prefix with as many letters as possible with the original word. We should not make any suggestions if the suggested word's prefix that matches the original word is not at least half the length of the original word. For example;

…
kane
karlina
karlinat
karlinqoyi
kartat
kartolat
kash
…

is a part from Dothraki vocabulary. If user searches the word "*karline*", the program should suggest the words "*karlina*", "*karlinat*" and "*karlinqoyi*" because these are the words with the maximum number of similar letters (in this case 6 letters) when we start our comparison from the first letter. Note that words like "*kartat*" are not offered since their matching prefix "*kar*" is not at least half the length of "*karline*".

You must use the AVL tree structure in order to make a decision (as many searches as needed) and to find the suggested words, if necessary. Other implementations that realize these operations will not be evaluated.

[1] Actually, the text file that you will use as vocabulary is a subset of the real Dothraki vocabulary. If you are interested, you can find the whole dictionary here: http://docs.dothraki.org/Dothraki.pdf

Based on this information, a sample interaction of the user with the program may be as follows:

```
Welcome to Dothraki Vocabulary!
This vocabulary contanins 378 words.
Enter a word: hador
  Found! The word is in vocabulary.
Enter a word: jerak
  Found! The word is in vocabulary.
Enter a word: yatha
  Not found!
  Some suggestions: yath yatholat
Enter a word: najahse
  Not found!
  Some suggestions: najah najahak najahat najaheya
Enter a word: bera
  Not found!
  Some suggestions: There is no word starting with letter b in vocabulary!
Enter a word: quit
Good Bye!
```

IMPORTANT NOTES

Please do not start the assignment before reading these notes.

- Before 23:55 on due date as specified above upload your solutions to the moodle.
- The name of the archive file must be in the following format.
  Surname_Name_StudentId_Section_HW4.zip
- You should upload a single zip file that contains
  - hw4.pdf, the file containing the answers to Q1.
  - .h and .cpp files containing the C++ source code. **Be sure to include the file with main method. Your files will be compiled with *g++ *.cpp* command and then it will be run with *./a.out* command. When your program works, user should be able to interact with the program as in the sample given in Q2.**

    - readme.txt, the file containing anything important on the compilation and execution of your program in Question 2.
    - Do not forget to put your name, student id and section number, in all of these files. Well comment your implementation.
- You are allowed to use the codes given in our textbook and/or our lecture slides. However, you ARE NOT ALLOWED to use any codes from somewhere else (e.g., from the internet, other text books, other slides ...).
- **IMPORTANT: Although you may use any platform and any operating system in implementing your algorithms and obtaining your experimental results, your code should work in a Linux environment with the g++ compiler. We will test your codes in a Linux environment. Thus, you may lose a significant amount of points, if your C++ code does not compile or execute in a Linux environment.**
- Keep all the files before you receive your grade.
- Be careful about indentation.
- Pay attention to these instructions, otherwise you may lose some points even though your code has no error.
- This homework will be graded by your TA, Hasan Balcı (hasan.balci at bilkent edu tr). Thus, you may ask your homework related questions directly to him.

DO THE HOMEWORK YOURSELF. PLAGIARISM AND CHEATING ARE HEAVILY PUNISHED!!!