

DATA STRUCTURES & ALGORITHMS 1

BATCH –A & B

[DEADLINE: APRIL 10TH 3:00 PM]

LAB ASSIGNMENT– 10

CODE:assign10

NOTES:

1. Please carefully read all assignments and there is **no choice**.
 2. **Follow variable and function naming conventions**
 - a. except for variables in for-loop, none of the other variables should be a single character.
 - b. The variable names and function names should indicate what they are storing/computing. For this assignment, we have given you some of the variable names and function names to use. They are highlighted as **function_name** or **variable_name**
 - c. All global variable should start with 'g_'
 3. Indentation improves readability. Please pick an indentation style and **indent your code** appropriately.
 4. Follow constants and type naming
 - a. All **constants** should be defined using **IFDEF and DEFINE**
 - b. All **structures** should have a **TYPDEF** to a simpler name
 5. When in doubt about naming or style conventions, consult the following link:
<https://users.ece.cmu.edu/~eno/coding/CCodingStandard.html>
 6. If your project contains multiple files, add all the files into a folder, zip and submit. Only the zip file should follow (rollno based) naming convention.
 7. **The file(s) you are showing to the instructors should match your submission. Even a minor modification (example: variable name change) is not acceptable.**
-

PROBLEMS [Total Marks: 20]:

You are going to practice how to implement a Trie and an AVL Tree.

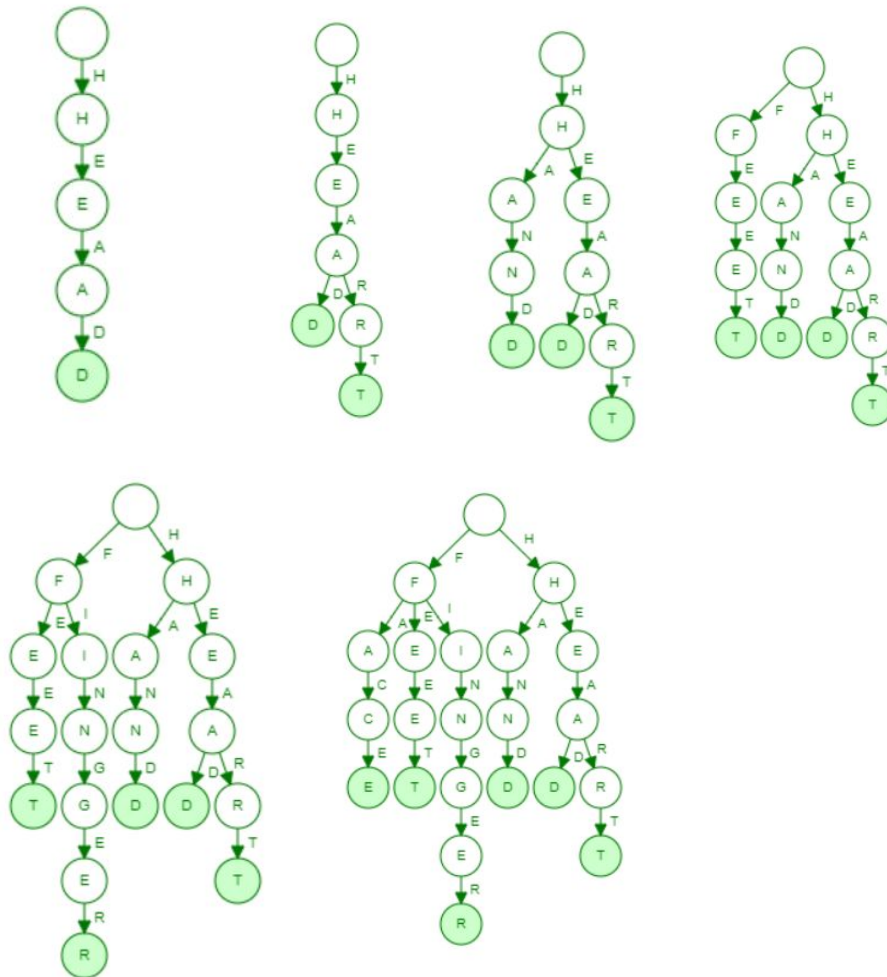
In computer science, a trie is an ordered multi-branch tree data structure that is used to store strings. Trie allows strings with similar character prefixes to share path (prefix). This way it reduces storage space and in addition can support a faster find/search capability. Lets see an example of maintaining a list of strings in a Trie.

String List

1. Head
2. Heart

3. Hand
4. Feet
5. Finger
6. Face

Let us look at what happens to the Trie after each insertion



Note: The colored nodes indicates that it completes a word. The node need not be a leaf.

Please try out more examples to get a feel for the trie:

<https://www.cs.usfca.edu/~galles/visualization/Trie.html>

Exercises:

1. [10 Marks] Implement a Trie to help search whether a given string is present in our list.
You need to implement the following:
 - a. addWord: Adds a word to the Trie
 - b. deleteWord: Deletes a word from the Trie
 - c. FindWord: Finds if the given word is in the Trie

2. *[10 Marks]* Implement a simple program which takes integers as input from the user and maintains an AVL Tree.
- a. Insert: Inserts the user entered integer into the tree
 - b. Delete: Deletes the user entered integer from the tree
 - c. Find: Finds the user entered integer if its present in the tree
 - d. (Obviously!) you need to write internal functions for different types of rotations

Note: The questions are simple for a reason. Take this assignment very seriously!