

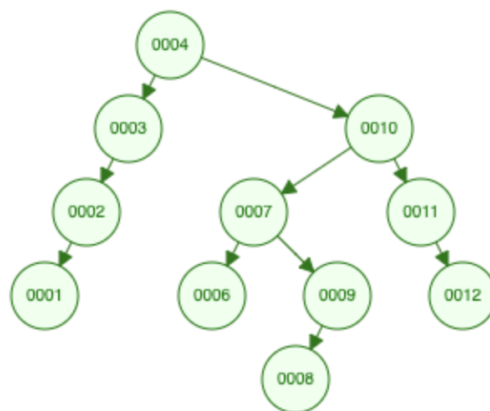
HW2

- Given the following letters and their frequencies, please provide the huffman encoding for each letter. (**Build the huffman encoding tree with smaller frequency on the left. Draw a separate tree for each in-between step**) Please submit your answer including drawing the tree on canvas.

A: 5
B: 9
C: 12
D: 13
E: 16
F: 45

AAAAABBBBBBBBBBCCCCCCCCCCCCDDDDDDDDDDDDDEEEEEEEEEEEEEEEFFF
FF

- Show the AVL tree that results after each of the integer keys 9, 27, 50, 15, 2, 21, and 36 are inserted, in that order, into an initially empty AVL tree. Clearly show the tree that results after each insertion, and clearly mark any rotations that must be performed.
- Rebalance the tree below. Draw a new tree for each rotation that occurs when rebalancing the AVL Tree. Please clearly mark the unbalanced node and the type of rotation you perform for each rotation.



- Please practice pre-order, in-order, and post-order tree traversal on Leetcode. You do not need to submit your code for this exercise..

<https://leetcode.com/explore/learn/card/data-structure-tree/134/traverse-a-tree/928/>
<https://leetcode.com/explore/learn/card/data-structure-tree/134/traverse-a-tree/929/>
<https://leetcode.com/explore/learn/card/data-structure-tree/134/traverse-a-tree/930/>

5. Create a new directory “HW2” in your github repo. For each of the following questions, create a single .cpp file with your code. (eg: 5a.cpp, 5b.cpp, 5c.cpp...) You may discuss with your peers. If you're stuck (after multiple attempts), you may look up online resources as long as you reference it, but you must NOT copy others' code. you MUST produce your own code. Please add proper comments in your code so that other people (and yourself!) can understand what the code is doing (even a few years later). Make sure your code passes all the test cases in leetcode . Finally, push your code under the HW2 folder.

- Please add proper comments in your code.
- Make sure your code passes all the test cases in leetcode .
- Push your code under the HW2 folder.
- Create a README.md file under the HW2 folder that shows how many test cases passed for each question. Eg:

<README.md>

- 5a - all pass
- 5b - all pass
- 5c - 10/202
- 5d - 2/80
- 5e - all pass
- ...
- ...

- a. [Level order traversal]

<https://leetcode.com/problems/populating-next-right-pointers-in-each-node/>

How does it know that it has reached the last treenode on each level and hence the next pointer should be set to null?

- b. [Traverse vs. subproblems]

<https://leetcode.com/problems/flatten-binary-tree-to-linked-list/>

Consider and compare both approaches, then implement either one.

- c. [Construct binary tree]

<https://leetcode.com/problems/construct-binary-tree-from-inorder-and-postorder-traversal/>

Please reflect on how we break it down to subproblems and solve it recursively.

[BST operations]

- d. <https://leetcode.com/problems/validate-binary-search-tree/>

- e. <https://leetcode.com/problems/insert-into-a-binary-search-tree/>
- f. <https://leetcode.com/problems/delete-node-in-a-bst/> (**Make sure you're actually moving the TreeNodes themselves, instead of swapping the int value of the TreeNodes.**)

[Trie]

- g. <https://leetcode.com/problems/implement-trie-prefix-tree/>

[Heap/priority queue]

- h. <https://leetcode.com/problems/k-closest-points-to-origin/>

Submission:

Please submit 1,2,3 on canvas, with a link to your github repo in the comment section.