CS101A(H) Homework 6

Due date: December 2, 2025, at 20:00

Your grade for this homework will be computed as

$$\min\big(\text{Single Choices} + \text{Multiple Choices} + \text{Others},\ 50\text{ pts}\big).$$

Notes: Unless otherwise stated, always express your final asymptotic bounds using $\Theta(\cdot)$ notation for tight complexity, not just $O(\cdot)$ or $\Omega(\cdot)$. All answers must be written **inside the provided answer boxes** and **in English**. When submitting, match your solutions to the problems correctly in Gradescope. No late submission will be accepted. Failure to follow these rule may result in partial or full loss of credit.

Usage of AI. The use of AI tools for searching information or obtaining assistance on this homework is strictly prohibited. All solutions must be derived from your own understanding and effort. Submissions will be reviewed carefully, and any indication of reliance on AI-generated content will result in severe penalties.

Academic Integrity. This course upholds the highest standards of academic integrity. Any form of academic dishonesty, including plagiarism, unauthorized collaboration, or the use of prohibited resources such as AI tools, will be treated as a serious violation. Such actions not only undermine your own learning but also violate university policies.

1. (10 points) Single Choice

(a) (2') Choose the *incorrect* statement about tree.

A. Nodes with the same parent are siblings

B. Leaf nodes always have a degree equal to 0.

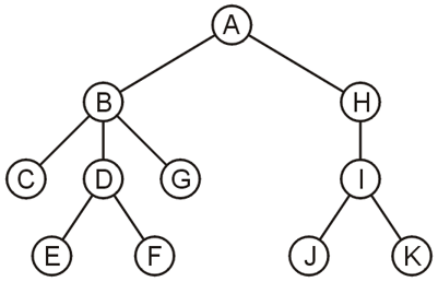
C. A tree of depth 2 has a maximum of 7 nodes.

D. Each node, other than the root, has exactly one node pointing to it.

(b) (2') Choose the correct statement.

A. We can use a stack to get the breadth-first traversal result of a tree.

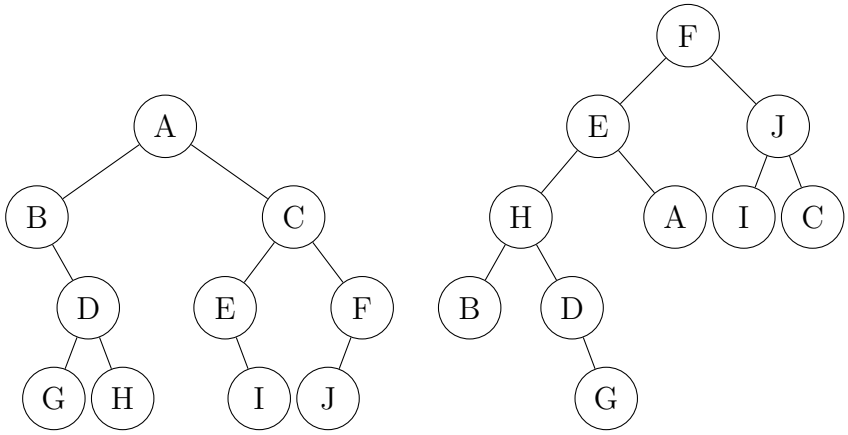
B. The BFS result of the figure below is ABHCDIGEFJK.



C. The memory required by DFS is $\Theta(n)$.

D. The time complexity of BFS and DFS are both $\Theta(n)$.

(c) (2') Which traversals of the left tree and right tree, will produce the same sequence node name?



A. left: Post-order, right: Pre-order

B. left: In-order, right: Pre-order

C. left: Post-order, right: In-order

D. left: In-order, right: Post-order

(d) (2') Choose the *incorrect* statement about Huffman coding.

A. Huffman coding prioritize characters based on their frequencies in text.

B. If character a has a higher frequency than b, then the encoded a has a length no longer than encoded b.

C. No code is a prefix of another code.

D. Given the set of characters and the **order** of their frequencies but the exact frequencies unknown, we can still determine the length of each encoded character.

(e) (2') Choose the correct statement about binary search tree.

A. Almost all of the relevant operations on a binary search tree are $O(h)$, where h is the height of the tree.

B. When deleting a node with two children, we first replace it with the smallest node on the right sub-tree or the smallest node on the left sub-tree.

C. Given an array, suppose we construct a BST (without balancing) by sequentially inserting the elements of the array into an empty BST. Then the time complexity of this process is $O(n \log n)$ in all cases.

D. The in-fix order traversal of a BST is an array of descending order.

2. (15 points) Multiple Choices

Each question has **one or more** correct answer(s). Select all the correct answer(s). For each question, you will get 0 points if you select one or more wrong answers, but you will get half points if you select a non-empty subset of the correct answers.

(a) (3') Choose the correct statement(s) about tree.

A. The length of a path is number of nodes in the path.

B. There is only one root in a tree.

C. The degree of a node is positive.

D. For every node in the tree, there is one parent node.

E. The height of a binary tree is always positive.

(b) (3') Choose the correct statement(s) about binary tree whose height is h, where h greater than 3.

A. Suppose it is a perfect binary tree with l leaf nodes, then this tree has $l-1$ internal nodes.

B. Suppose it is a complete tree whose left sub-tree has n nodes, then the range of n is between $2^{h-1}+1$ and 2^h.

C. Suppose it is a full binary tree, the number of nodes is always no less than 2^h.

D. If the result of in-order traversal of an expression tree is $3\times4\times5+1+2+5\div5+6-7$, then the result of the expression represented by this tree can be 63.

(c) (3') Choose the correct statement about heap using complete trees.

A. In the worst case, the time complexity of `Push` operation is $\Theta(n)$ if we do not use complete tree to implement a heap.

B. Usually `Push` operation takes $O(\log n)$ time.

C. Heap sort with a max-heap takes $\Theta(n\log n)$ time and is an in-place sorting algorithm.

D. Merging two binary heaps of size n is a $\Theta(n)$ operation

(d) (3') Consider an AVL tree whose number of nodes is n and height is h, which of the following are true?

A. Inserting or removing a node can increase the height of a tree by at most 1.

B. An insertion on a perfect binary tree (which is also an AVL tree) can cause an imbalance.

C. h may equals to $\Theta(n)$ in worst case.

D. $n = O(2^h)$

(e) (3') Choose the correct statement about binary search tree.

A. In a BST, the nodes in a subtree appear contiguously in the in-order traversal sequence of the BST.

B. If we erase a node that has two children, then it can be replaced by the maximum object in its right subtree.

C. There are 5 different BSTs of a set with 3 distinct numbers.

D. If a node doesn't have a right subtree, then its **next** (defined in lectures) object is the largest object (if any) that exists in the path from the node to the root.

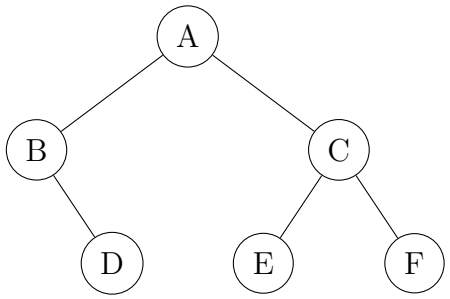
3. (10 points) Run DFS and BFS

Answer the following questions for the tree shown below **according to the definition specified in the lecture slides**.

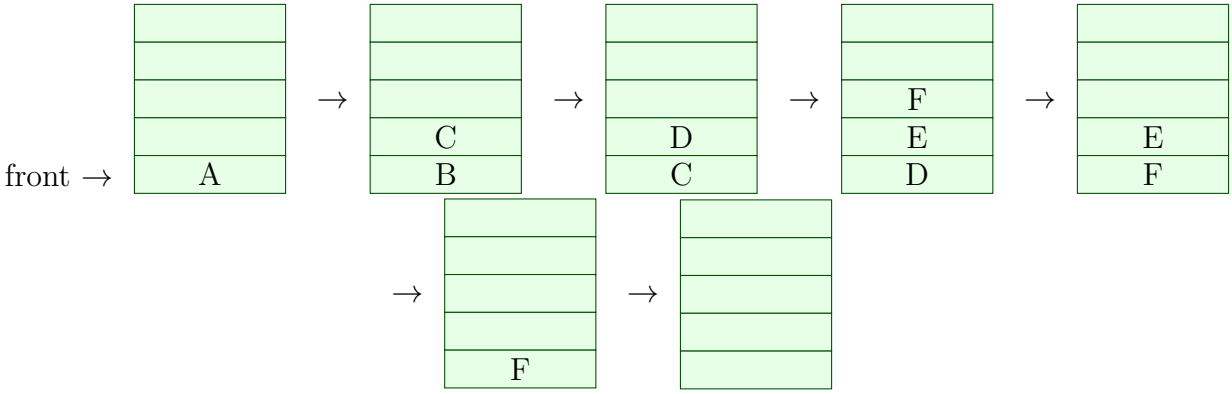
Note: Form your answer in the following steps.

1. Decide on an appropriate **data structure** to implement the traversal.
2. **Popping a node** and **pushing a sequence of children** can be considered as one single step.
3. When doing **Breadth First Traversal**, push children of a node into the data structure in **alphabetical order**; when doing **Depth First Traversal**, push children of a node into the data structure in **reverse alphabetical order**.

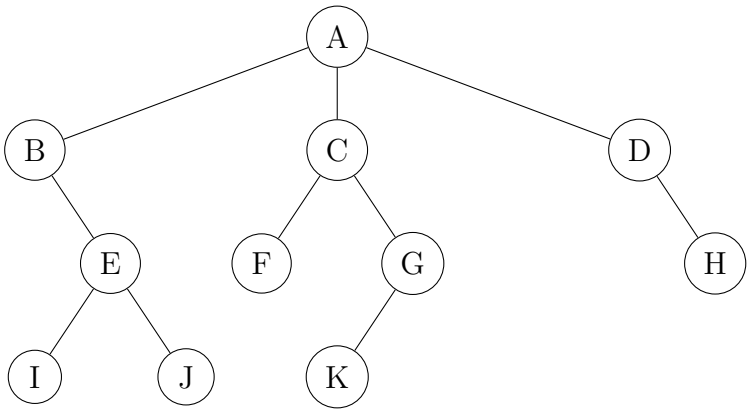
Example: Given a tree with root **A**:

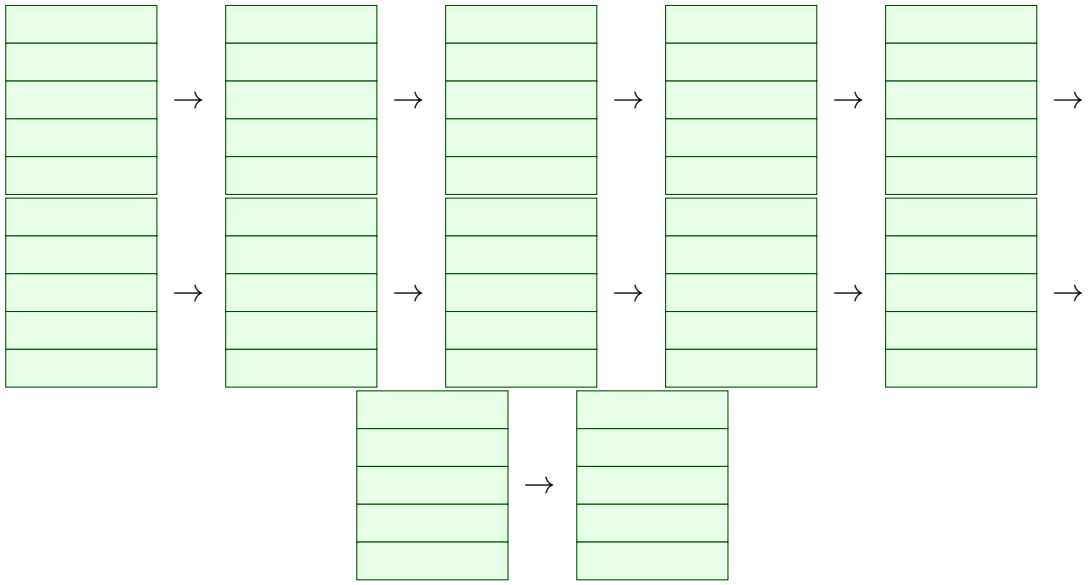


The process of doing **Breadth First Traversal** is:

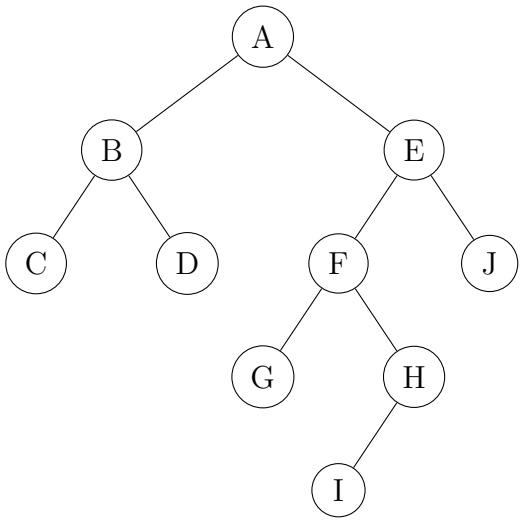


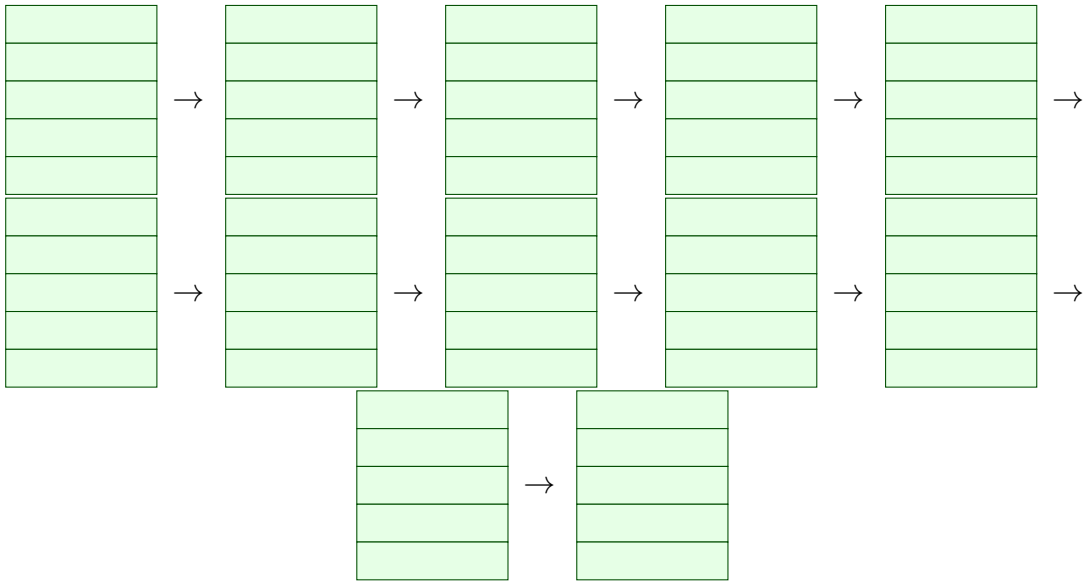
(a) (5') Run **Pre-order Depth First Traversal** on the tree with root **A** and draw the whole process in the space below.





(b) (5') Run **Breadth First Traversal** on the tree with root **A** and draw the whole process in the space below.





4. (8 points) Draw a binary tree

(a) (3') Given the in-order and pre-order traversal of a binary tree T are IGDHBAECF and ABDGIHCEF respectively.
Draw the tree T.

(b) (3') Given the in-order and post-order traversal of a binary tree T are ADHGKLMRUXTW and AHDLKGUXRWTM respectively.
Draw the tree T.

(c) (2') Given the pre-order and post-order traversal of a binary tree T, can you decide the tree T? If yes, please describe an algorithm to construct T; if no, please provide a counterexample.

5. (6 points) Huffman Coding

After you compress a text file using Huffman Coding Algorithm, you accidentally spilled some ink on it and you found that one word becomes unrecognizable. Now, you need to recover that word given the following information:

Huffman-Encoded sequence of that word:
00001100111101
Frequency table that stores the frequency of some characters:

characters	b	e	i	k	m	t	x
frequency	12	9	6	7	2	1	5

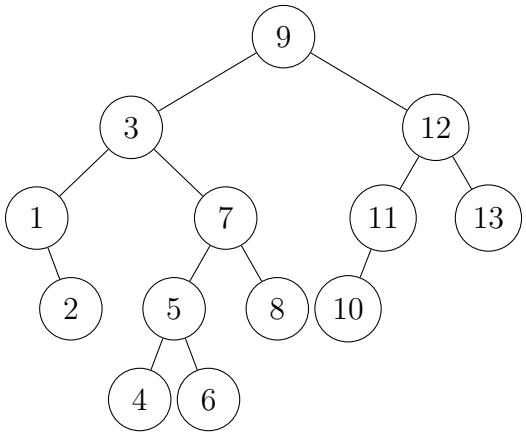
(a) (4') Please construct the binary Huffman Coding Tree according to the given frequency table and draw the final tree below.
Note: The initial priority queue is given as below. When popping nodes out of the priority queue, the nodes with the same frequency follows "First In First Out".

t	m	x	i	k	e	b
1	2	5	6	7	9	12

(b) (2') Now you can "decompress" the encoded sequence and recover the original word you lost. Please write the original word below.

6. (8 points) AVL tree operations

Here is an AVL tree. Denote it as T.



(a) (2') Insert 5.5 into T. Draw the AVL tree before checking if any balance correction is needed.

(b) (2') Insert 5.5 into T. Draw the AVL tree after balance corrections.

(c) (2') Remove 12 from T (**NOT from the previous answer!**). Draw the AVL tree after replacing and before checking if any balance correction is needed.

Note: when erasing a non-leaf node x, we will follow the way in BST lecture slides to replace x with

- the child of x, if x has exactly one child, or
- the successor of x, if x has two children

(d) (2') Remove 12 from T. Draw the AVL tree after balance corrections.