CS101 Algorithms and Data Structures
Fall 2023
Homework 5

Due date: November 12, 2023, at 23:59

1. Please write your solutions in English.
2. Submit your solutions to gradescope.com.
3. Set your FULL name to your Chinese name and your STUDENT ID correctly in Account Settings.
4. If you want to submit a handwritten version, scan it clearly. `CamScanner` is recommended.
5. When submitting, match your solutions to the problems correctly.
6. No late submission will be accepted.
7. Violations to any of the above may result in zero points.

1. (12 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 1 point if you select a non-empty subset of the correct answers.

Write your answers in the following table.

(a)	(b)	(c)	(d)	(e)	(f)
AC	ABD	A	B	A	ABD

(a) (2') Which of the following statements is/are true about **tree**?

A. A tree with N nodes has $N - 1$ edges.

B. The height of a tree is always positive.

C. Every node of a tree is either a leaf node or an internal node.

D. The degree and the depth of the root node should be 0.

(b) (2') Which of the following statements is/are true about **binary tree**?

A. In a binary tree, every non-root node has exactly one parent.

B. Every full binary tree is also a complete binary tree.

C. A full binary tree with n non-leaf nodes contains $2n - 1$ total nodes.

D. A binary tree of height 0 is also perfect.

(c) (2') Which of the following choices is/are $\Theta(m)$ where m is the maximum length of the queue when traversing a tree with BFS?

A. The total number of nodes in the tree.

B. The length of the deepest path from a leaf node to the root node.

C. The maximum degree of nodes in the tree.

D. The maximum number of nodes at a given depth of the tree.

(d) (2') There exists two paths between any two different nodes in a tree with height more than 3.

A. True.

B. False.

(e) (2') The height of a tree is always equal to the maximum depth of nodes in the tree.

A. True.

B. False.

(f) (2') Which of the following statements is/are false?

A. Nodes with the same depth are siblings.

B. Each node in the tree has exactly one parent pointing to it.

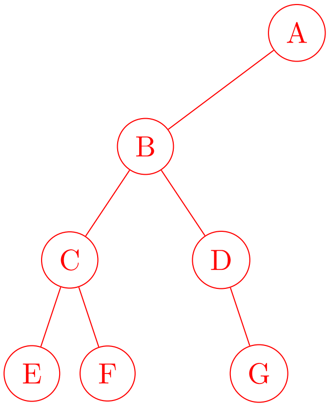
C. Given any node α within a tree, the collection of α and all of its descendants is a subtree of the tree with root α.

D. The root node cannot be the descendent of any nodes.

2. (10 points) Generate a binary tree

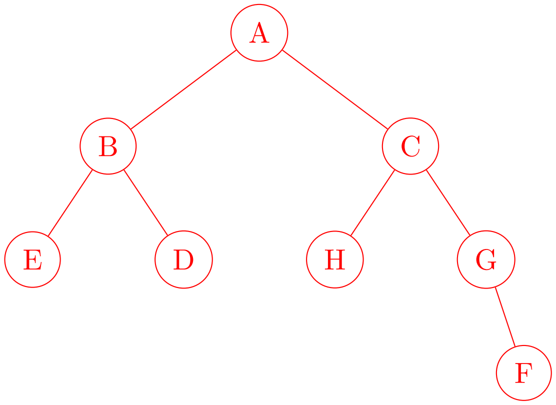
(a) (4') Given the in-order and pre-order traversal of a binary tree T are **ECFBDGA** and **ABCEFDG** respectively.
Draw the tree T.

Solution:



(b) (4') Given the in-order and post-order traversal of a binary tree T are **EBDAHCGF** and **EDBHFGCA** respectively.
Draw the tree T.

Solution:



(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.

Solution:
Knowing the pre-order and the post-order, we can construct the tree.

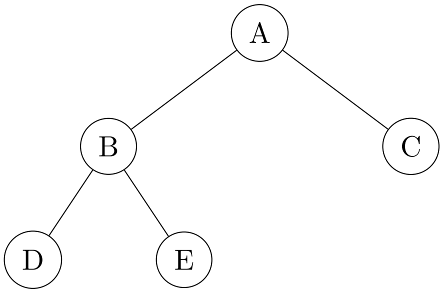
1. The first element in the pre-order and the last element in the post-order is the same, which is the root. Remove it.
2. The second element in the pre-order is the first children of the first element, thus in the post-order elements in front of it form a subtree.
3. Regard the subtree as a new tree and repeat the operations above.
4. If facing with the elements shares the same position in pre-order and post-order, this element is a leaf. For example, if ACB is the pre-order and ABC is the post-order, A will be the leaf of the tree. Remove it.

3. (10 points) Tree Structure and Traversal

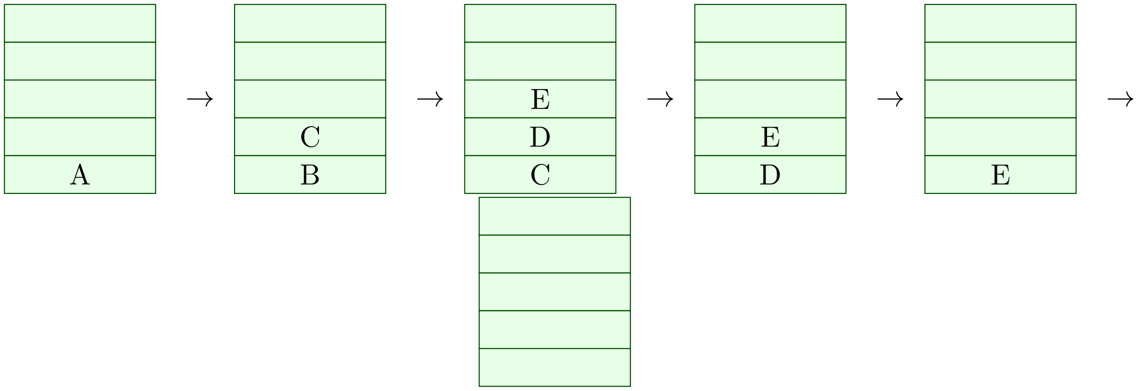
You should solve the below questions following these steps:

1. Decide on an appropriate **data structure** to implement the traversal.
2. When doing **Breadth First Traversal**, push children of a node into the data structure in alphabetical order.
3. Consider **poping an entry** and **pushing all its children** as one step.

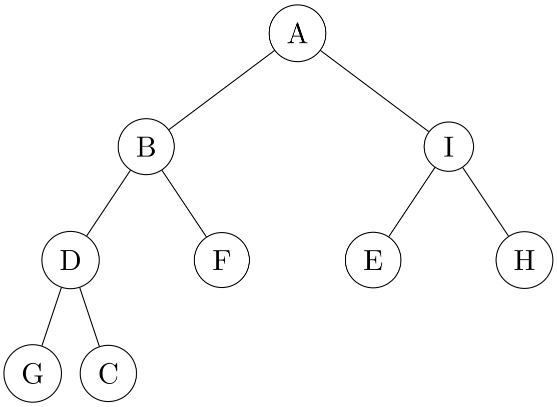
Example: Given a tree with root **A**:



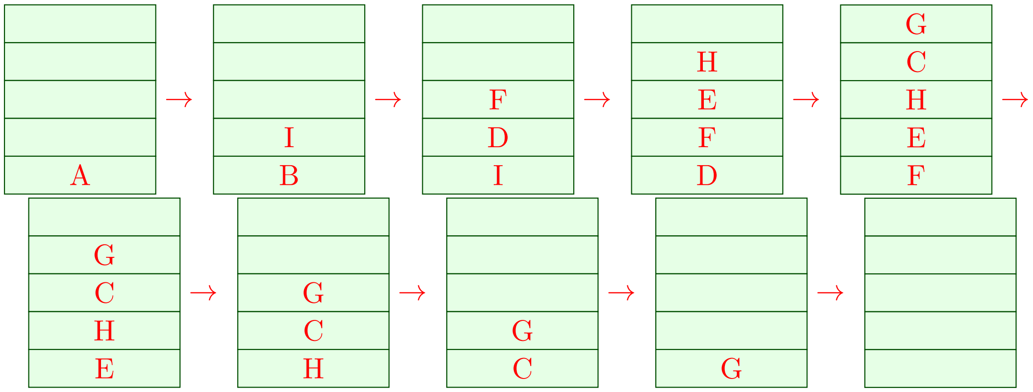
The process of doing **Breadth First Traversal** is:



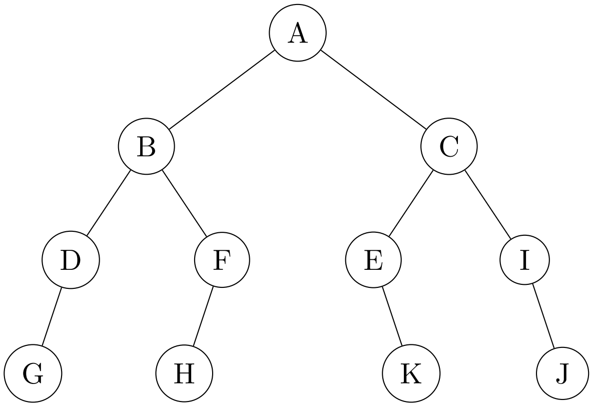
(a) (5') Run **Breadth First Traversal** on the tree with root **A** and draw the whole process in the space below.



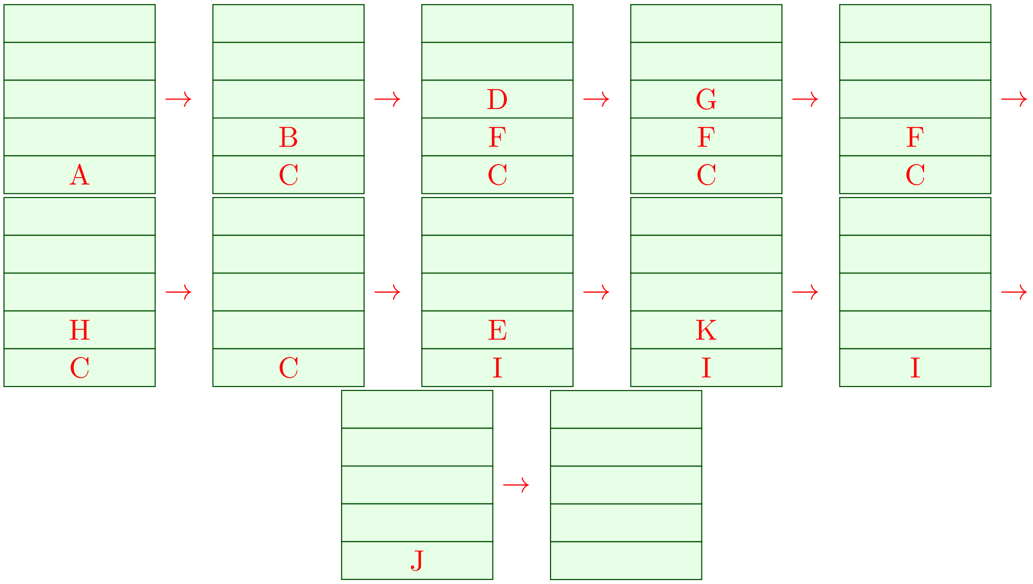
Solution:
Breadth First Traversal can be implemented by queue. The process of doing **Breadth First Traversal** is:



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



Solution: It can be implemented by stack. The process of doing **Pre-order Depth First Traversal** is:



4. (10 points) Left Child Right Sibling

(a) (1’) For every ordered tree, there is a unique representation of Left-child right-sibling format.

√ True ○ False

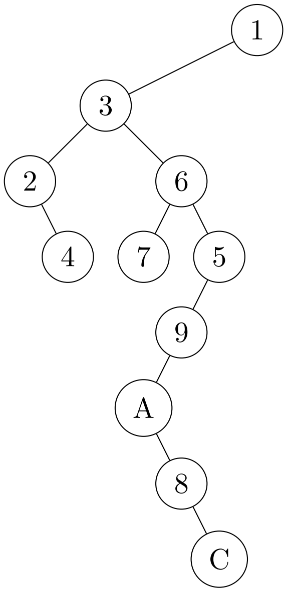
(b) (1’) Pre-order traversal of the original tree is identical to the pre-order traversal of the Knuth transform.

√ True ○ False

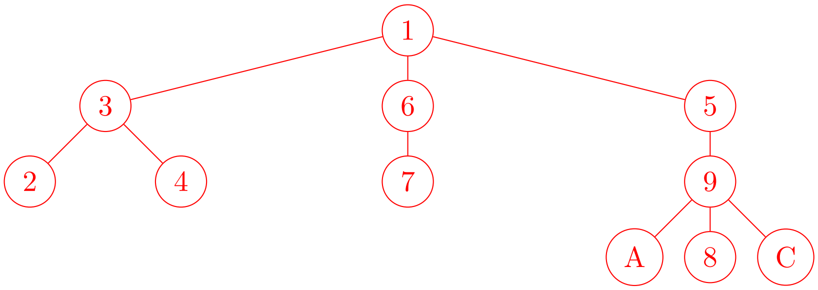
(c) (1’) Post-order traversal of the original tree is identical to the post-order traversal of the Knuth transform.

○ True √ False

(d) (7’) Transform the tree below with root **1** (in LCRS format) to N-ary format.



Solution:


1
3
6
5
2
4
7
9
A
8
C