

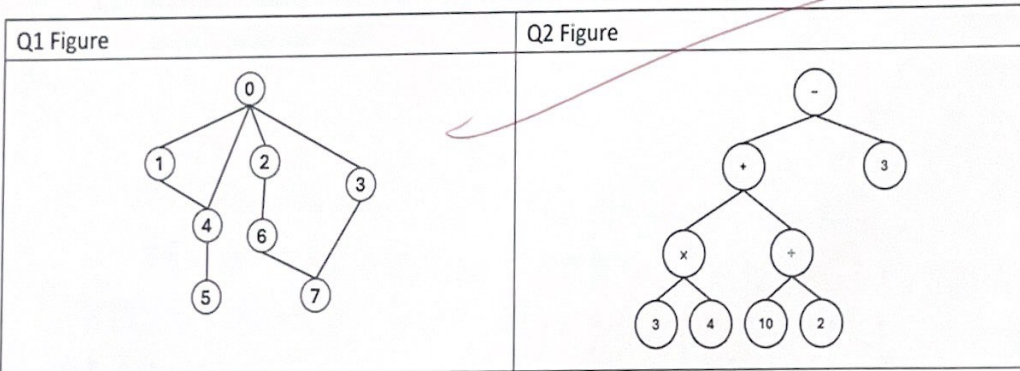
Course: Data Structures (CSE CS203A, 114-1)  
Take-Home Quiz IV: Tree/Heap/Graph

Due: December 16, 2025, 17:00 (Room R1102)

Important Notice: You must print this take-home quiz and write your answers by hand with a pen.

Student ID: 1133322

Student Name: 林百達



Q1. (30 pts) Explain Breadth-First Search (BFS) on the graph and provide the BFS traversal order for the graph shown in Q1 Figure.

A1: ① 廣度優先搜尋 (Breadth-First Search, BFS) 是一種圖形或樹的遍歷演算法，自指定的起始節點出發，先拜訪距離最近的節點，再逐層擴展至較遠的節點，以完整探索所有可到達的節點，特別適用於尋找無權重圖中的最短路徑。BFS 透過佇列 (Queue) 以先出 (FIFO) 的方式管理節點的處理順序，確保遍歷過程能依照層級由近至遠的順序進行。

- ② (1) 選定起始節點並初始化：起始節點 0；queue = [0]；visited = {0}
- (2) 取出節點 0，加入鄰居 1, 2, 3, 4；queue = [1, 2, 3, 4]；visited = {0, 1, 2, 3, 4}
- (3) 取出節點 1，(鄰居皆已訪問)；queue = [2, 3, 4]；visited = {0, 1, 2, 3, 4, 6}
- (4) 取出節點 2，加入鄰居 6；queue = [3, 4, 6]；visited = {0, 1, 2, 3, 4, 6, 7}
- (5) 取出節點 3，加入鄰居 7；queue = [4, 6, 7]；visited = {0, 1, 2, 3, 4, 5, 6, 7}
- (6) 取出節點 4，加入鄰居 5；queue = [6, 7, 5]；visited = {0, 1, 2, 3, 4, 5, 6, 7}
- (7) 取出節點 6，(鄰居皆已訪問)；queue = [7, 5]；visited = {0, 1, 2, 3, 4, 5, 6, 7}
- (8) 取出節點 7，(鄰居皆已訪問)；queue = [5]；visited = {0, 1, 2, 3, 4, 5, 6, 7}
- (9) 取出節點 5，(鄰居皆已訪問)；queue = []；visited = {0, 1, 2, 3, 4, 5, 6, 7}

BFS Traversal Order: 0, 1, 2, 3, 4, 6, 7, 5

Q2. (30 pts) In tree traversal, one common method is inorder traversal. Please use inorder traversal to print the arithmetic expression represented by the expression tree in Q2 Figure, and then evaluate it to compute the final result.

過程	輸出
(1) 走到根節點「-」，依中序規則先走左邊，進入「+」	
(2) 進入「+」，依中序規則先走左子樹，進入「x」	
(3) 進入「x」，依中序規則先走左子樹，進入 3 (葉節點)，故直接輸出 3	3
(4) 因左子樹已完成，回到「x」，依中序規則輸出根節點「x」	3x
(5) 走「x」的右子樹，進入 4 (葉節點)，故直接輸出 4	3x4
(6) 回到「+」，依中序規則輸出根節點「+」	3x4+
(7) 走「+」的右子樹，進入「÷」	
(8) 在「÷」，依中序規則先走左，進入 10 (葉節點)，故直接輸出 10	3x4+10
(9) 因左子樹已完成，回到「÷」，依中序規則輸出根節點「÷」	3x4+10÷
(10) 走「÷」的右子樹，進入 2 (葉節點)，故直接輸出 2	3x4+10÷2
(11) 回到整棵樹的根節點「-」(左子樹已完成)，依中序規則輸出根節點「-」	3x4+10÷2-
(12) 走右子樹 (葉節點 3)，葉節點直接輸出 3	3x4+10÷2-3

計算結果：3x4+10÷2-3 = 14

Q3. (40 pts) A binary tree is a fascinating data structure with many variations, including binary search trees, AVL trees, red-black trees, complete binary trees, and max/min heaps. These variations can be classified as shape-based (structural constraints) or criteria-based (rules such as ordering). Choose one shape-based tree and one criteria-based tree, and provide a brief description of each.

A3:

1. 依外形限制的樹 (Shape-based Tree): 完全二元樹 (Complete Binary Tree)

完全二元樹是一種二元樹，其中除了最底層之外的所有層級必須被填滿，而最底層的節點必須自左而右連續排列

2. 依規則限制的樹 (Criteria-based Tree): 二元搜尋樹 (Binary Search Tree, BST)

二元搜尋是一種二元樹，節點須滿足：左子節點的值小於根節點，而右子節點的值大於根節點。此種排序規則使得搜尋、插入與刪除能有效率地進行，平均時間複雜度可達  $O(\log n)$