Session 24
Strategy 3: Divide-and-Conquer

Lecture Summary

Focus on illustrating the algorithm design strategy and applying analysis techniques. Students have studied to death how binary trees work in previous courses.

Binary Trees

1. Review definitions and terminology from 🕮 1.4
2. Typical binary tree divide-conquer algorithm: tree height
3. Performance of divide-conquer tree height and similar binary tree algorithms fitting the pattern
4. Compare to processing pattern of insert in a binary *search* tree

Session Exercise

> 👁 *The point of the following exercises is to write the pseudocode first then code to test its correctness. Use the binary tree starter code from the course assignment page.*

P17. Design algorithms for the following: counting leaves, counting nodes, and the 3 main node traversals.

P18. Complete the binary tree object as detailed in the code starter file. Can a preorder and a postorder sequences specify a binary tree uniquely?

P19. Use the code to test your solution of 🕮 Exercise 5.3:5.

> **Hint**: use your solution as input lists, browse the resulting tree in the console to ensure it is created correctly (should match the figure). Finally, output the tree traversals from your methods (should match your original answers). Also try 5.3:8a to see if you can match the preorder list.

🕮 **Exercise 5.3** • 2, 4, 5, 6 🛠 1, 8c, *11

Reading List

- 🕮 5.3
- 🕮 1.4 (pp. 33-34)

Keywords

Node depth, [Tree] height, [Tree] Traversal