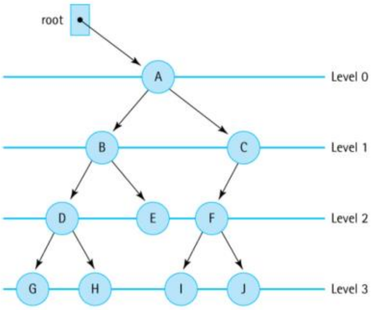
CSC 241

Lab 7

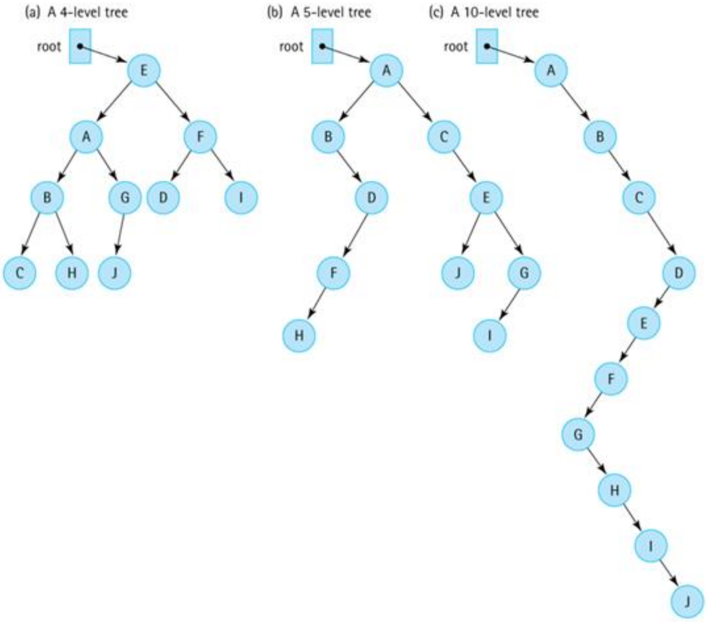
Option 1:

Fullness Experiment:

a. Design and implement a method `height` for `BinarySearchTree` that returns the height of the tree.
b. Define the fullness ratio of a binary tree to be the ratio between its minimum height and its height (given the number of nodes in the tree). For example, the following tree:



has a fullness ratio of 1.00 (its minimum height is 3 and its height is 3) and the following 10-level tree:


(a) A 4-level tree

(b) A 5-level tree

(c) A 10-level tree

has a fullness ratio of 0.33 (its minimum height is 3 and its height is 9). Implement a method `fRatio` to be added to the `BinarySearchTree` class that returns the fullness ratio of the tree.

c. Create an application that generates 10 “random” trees, each with 1,000 nodes (each node contains a random integer between 1 and 3,000). For each tree, output its height, minimum height, and fullness ratio. Recall, the optimal height h of a binary tree is $\lfloor log_2 N \rfloor$. For example, a tree with 20 nodes has an optimal height: $h = \lfloor log_2 20 \rfloor = \lfloor 4.32192809 \rfloor = 4$.
d. Submit a report that discusses how the `fRatio` method might be used by an application to keep its search trees reasonably well balanced.

Option 2:

Word Frequency Applications:

Use the WordFreq class file provided in the source to complete the following:

Create an application that will read a text file (.txt) and:

a. Display the longest word (or words if there is a tie) in the file and how many times it occurs.
b. Display the most frequently used word (or words if there is a tie) in the file and how many times they occur.
c. Display the word or words in the file that occur exactly once.