# Week 3 - Hands On Exercise - Review Questions

ASD103A-21: Object-Oriented Data Structures using Python, Part1

Prof: Debsankar Mukhopadhyay

Delvis Soto-Soto

Software Developer

Question 1: What's the diff between a perfectly balanced binary tree and a complete one?

A perfectly balanced binary tree keeps things chill by making sure the heights of its left and right child subtrees don't differ by more than one. It's like keeping everything balanced and in check.

Now, a complete binary tree is like that friend who likes to fill seats from the left. It wants all levels full, except maybe the last one, and it's not too fussed if it's not perfectly filled.

Question 2: How 'bout a complete binary tree vs. a full one?

Well, a complete binary tree is all about being as left as possible. It fills levels from left to right but isn't too strict about filling up the last level.

A full binary tree is simpler – every node's either got no kids or two kids. It's that binary simplicity, no in-betweens.

Question 3: How many nodes are kickin' in a full binary tree with a height of 5?

Time to count:

25+1−1=64−1=63 , 25+1 −1=64−1=63. So, our full binary tree with height 5 has 63 nodes.

Question 4: What's the deal with a binary search tree compared to a regular binary tree?

A binary tree is like a general hangout – no rules, just nodes and connections. Now, a binary search tree? It's got some manners. Every node has a value, and if it's chillin' to the left, it's smaller, and if it's on the right, it's bigger. Makes searching for stuff way easier!