

1. What does it mean if a binary search tree is a *balanced tree*?

Balanced tree means the difference of heights in left and right subtrees of any node is within 1. If a binary search tree is a balanced tree, that means the tree is sorted and subtrees are also balanced. The first balanced binary search tree is also called AVL tree.

2. What is the big-Oh search time for a balanced binary tree? Give a logical argument for your response. You may assume that the binary tree is perfectly balanced and full.

The time complexity for balanced binary tree is  $O(\log N)$ . Because the binary tree is balanced, depth of the tree will be  $O(\log N)$ . For example, if we have 7 nodes, in the balanced binary tree, there will be 3 levels. In the worst case, the value we are looking is in the bottom level of the tree, and we will need to go through every level to find it. Therefore, the big-Oh search time for balanced binary tree is  $O(\log N)$ .

3. Now think about a binary search tree in general, and that it is basically a linked list with up to two paths from each node. Could a binary tree ever exhibit an  $O(n)$  worst-case search time? Explain why or why not. It may be helpful to think of an example of operations that could exhibit worst-case behavior if you believe it is so.

In the worst case, the search time for binary tree could be  $O(n)$ . When the tree is not balance and only have one subtree for each node, the search time will be same as the linked list which is  $O(n)$ . For example, if the root is 1 and the right child is 2, and the right child for 2 is 3. The tree is a linked list in this way.