 

A binary tree node has data, pointer to left child and a pointer to right child. root is placed at the top. Two child nodes are merging into root. and each node is from the right and left, and two nodes are merging. For the binary tree to be balanced, the lengths of the nodes located on the left from the root, together with the lengths of the nodes located on the right, must be one node more or one node less. when this difference is larger, the binary tree is not balanced. I wrote a code that checks for the height balance of the binary tree. In the code I wrote, I get the child nodes located on the left from the root and the child nodes located on the right from the root. If there is no other child node on the left after the node located on the left, the code looks at the child node located on the right. Likewise, if the node located on the right does not have a child node located on the right after it, the code looks at the child node located on the left. I include nodes in the code according to their location. If the answer is different from 1.0 or -1 when we subtract the lengths of the child nodes located to the right from root from the length of the child nodes located on the left from root, this binary tree is not balanced. If the answer is with 1, 0, or -1, then this node is balanced.