

Question 3

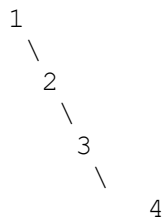
Self-balancing binary search tree

We know that binary search trees run all operations add, delete, get, and nearest in $O(n)$. Where n is at least the height of the tree.

Further more, if we maintain a balanced tree, then those operations are no longer $O(n)$ because the maximum traversal is still the height of the tree, but now the tree balances, so the height of the tree is at most $\log_2(n)$. Therefore we conclude that a balancing binary search tree is $O(\log n)$.

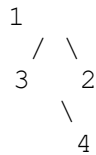
Example: $N = 4$, $\log_2(N) = 2$

Binary Search Tree worst case scenario:



Time (= height): $O(4)$

Balanced Binary Search Tree:



Time (= height): $O(2)$