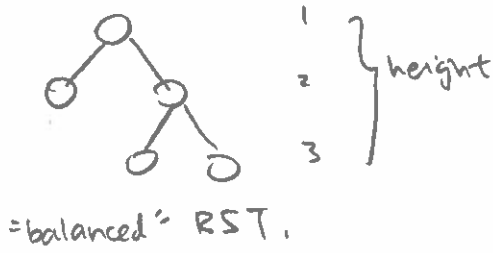


# Kevin's BST write up

\*this is a simple note/observations regarding BSTs.

## Height balancing Tree:



For binary search trees, search time is best when the tree is not degenerate i.e when the height level between leaves is not greater than one.

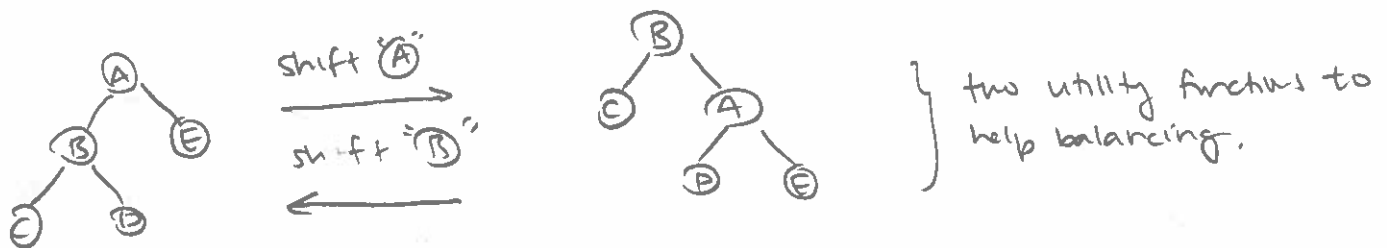
Here, we can observe a balanced tree, where the difference is = 1 from level 2 and level 3 heights.

Traditional BST will have a worst case efficiency of  $O(n)$ , when the tree is degenerate.

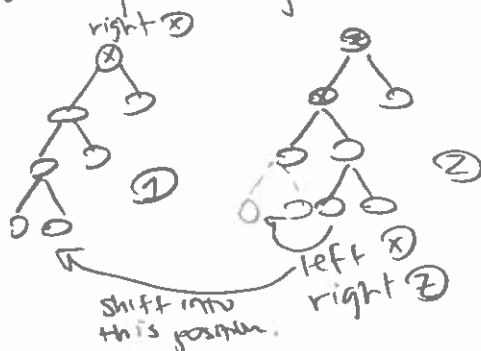


However, when balanced, the efficiency will decrease to  $O(\log n)$ . Because of the even distributions.

The algorithm for balancing trees is relatively intuitive. We can approach each `put()` and `delete()` call with a correction after to balance the height. For instance:



Each <sup>sub</sup>tree can be broken down into 4 types assuming the tree was previously balanced:



③ & ④ are minors.

For these trees, we can balance them by 1 or 2 shifts to balance them as such.

