Problem: Given a binary tree, determine if it is valid binary search tree.

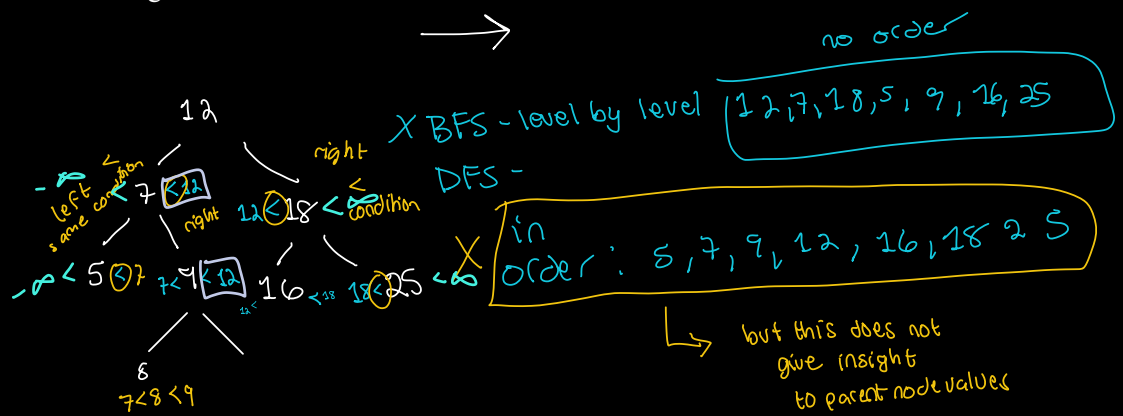
Intro:
- Verify Constraints
- Create Testcases

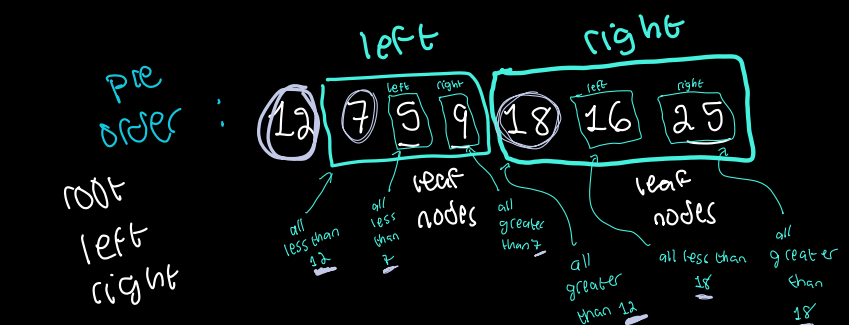
|||

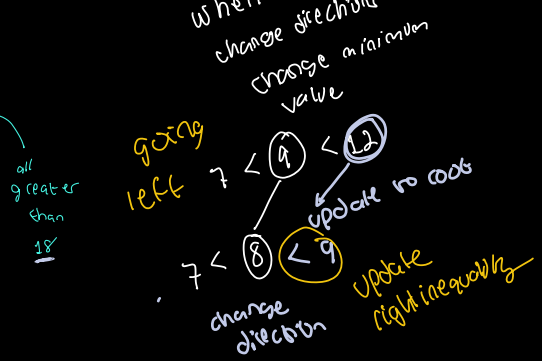
Brute Force:
- Brainstorming & Pattern Observations
- Pseudocode
- Write code
- Run through testcases
- Analyze time and space complexity

→ in Order traversal
Time: O(n)
space: O(1)

Optimal:
- Brainstorming & Pattern Observations







How to handle boundaries?

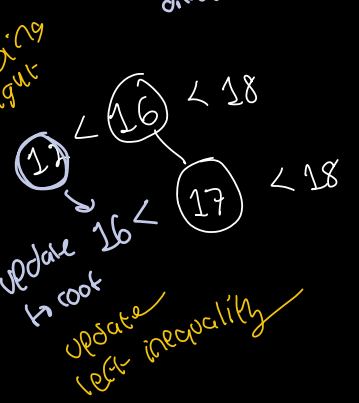
left most
$-\infty < 5$

right most
$25 < \infty$

=> what are the constraints

Node values:
$-2^{31} \le$ Node.val $\le 2^{31} - 1$

$-\infty$ $-2^{31}-1$ ∞ 2^{31}



- Pseudocode
- Write code
- Run through testcases
- Analyze time and space complexity
    - Time: O(n)
    - Space: O(n)

worst case
touch all the nodes

but avg. case much better than brute force