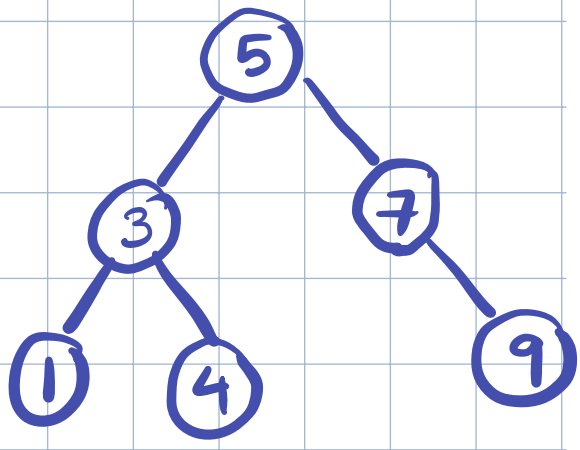
TUT WEEK04 — Binary Search Trees

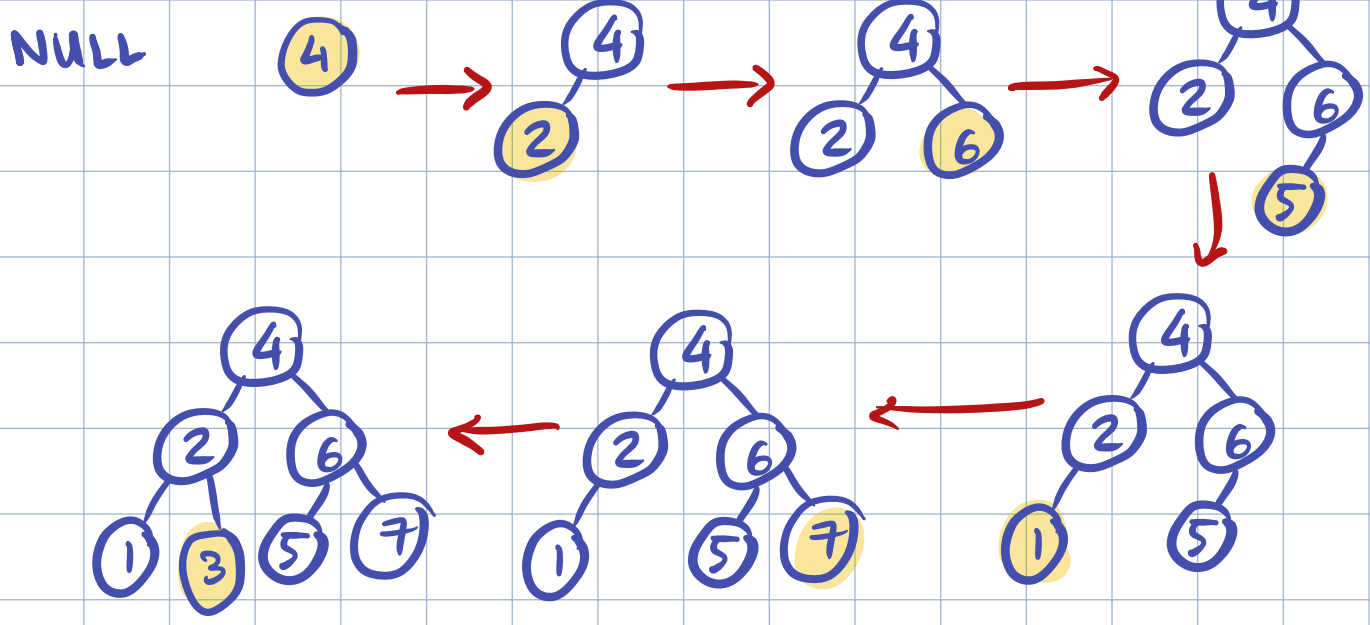
- Consists of Nodes
  - value
  - left child
  - right child

```
Struct BSTNode {
    int value;
    struct BSTNode left;
    struct BSTNode right;
};
```

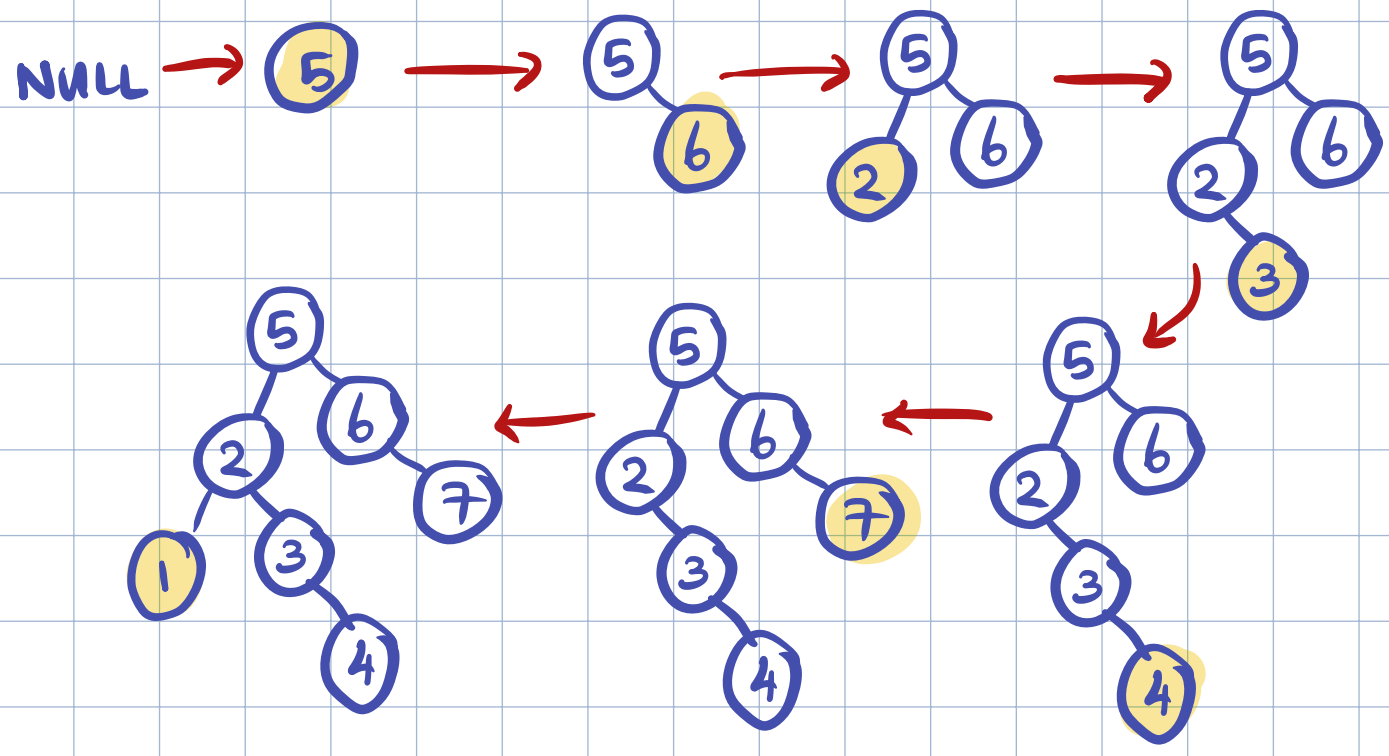


- Trees can only have up to 2 children
  - (hence 'Binary')
- Trees are a recursive data structure
  - if you look at (3), it is like its own smaller tree
  - this is why recursive functions work well with trees!
- Nodes also have to be in order
  - left child < curr < right child

Q1. if we added nodes 4 2 6 5 1 7 3 to an empty tree ... (adding as a leaf)

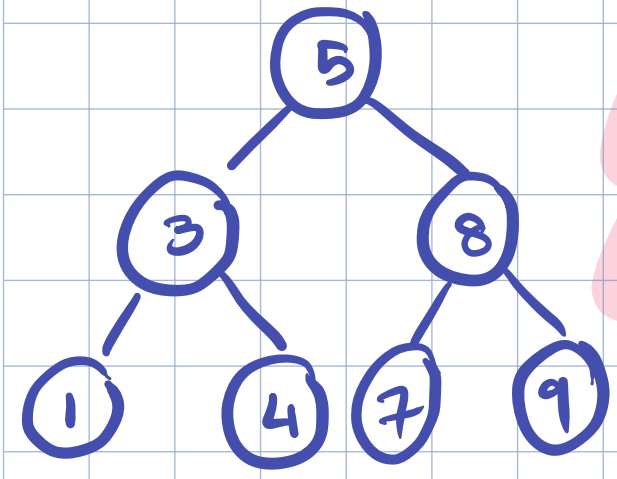


but if we add the same numbers in a different order : 5 6 2 3 4 7 1



* notice how we get a different tree depending on the order!

Q2. Tree Order



- Infix : left, centre, right
- Prefix : centre, left, right
- Postfix : left, right, centre
- Level Order : Top to Bottom

Infix(5) = 1 3 4 5 7 8 9

↳ even though we start at 5, and go 'left, middle, right', instead of printing '3' first, we have to print infix(3), which is '1, 3, 4'

Prefix(5) = 5 3 1 4 8 7 9

Postfix(5) = 1 4 3 7 9 8 5

Level Order(5) = 5 3 8 1 4 7 9