Quiz 04 Review Session !

COMP 210 / 2024 Summer Session I

Ajay Gandecha

Quiz 04 Format

- 30 minutes at the start of class.
- *On paper* - bring a pencil!
- **Question Types:**
  - Multiple choice, T/F, select all that apply, fill in the blank, drawing trees

Exercise Check-In Question

- Similar format to the exercise question on Quiz 02 and Quiz03.
- Review **`Ex06`** and **`Ex07`**.
- *Questions?*

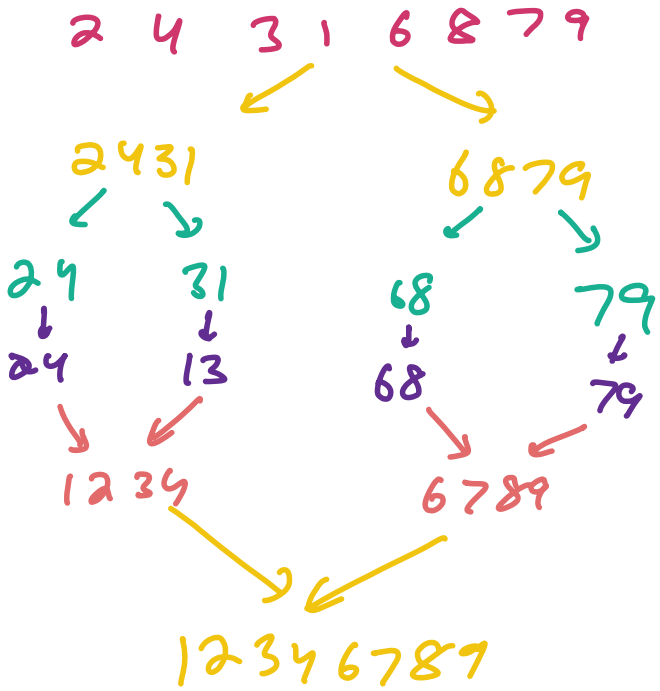
On Quiz 04

Time Complexity of Sorting (Quick sort, bubble sort) - Both avg. and worst case

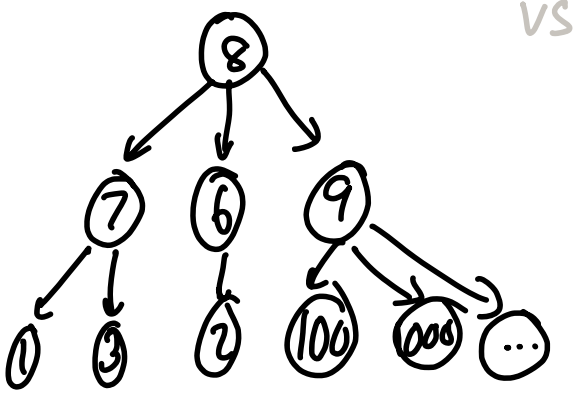
- Trees and Binary Trees
- Binary Search Trees and its “Invariants” (Guaranteed properties)

Sorting Analysis

- **Quick Sort**
  - Average Case: O(NlogN)
  - Worst Case: $O(N^2)$
- **Merge Sort**
  - Average + Worst Case: O(NlogN)
- **Bubble Sort**
  - Average + Worst Case: $O(N^2)$

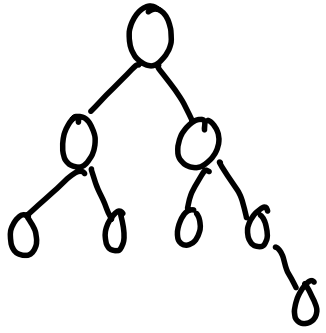


Trees



vs

Binary Tree

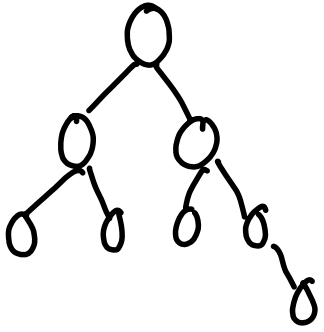


Only max of two branches from any node allowed

vs

(Binary Search Tree)

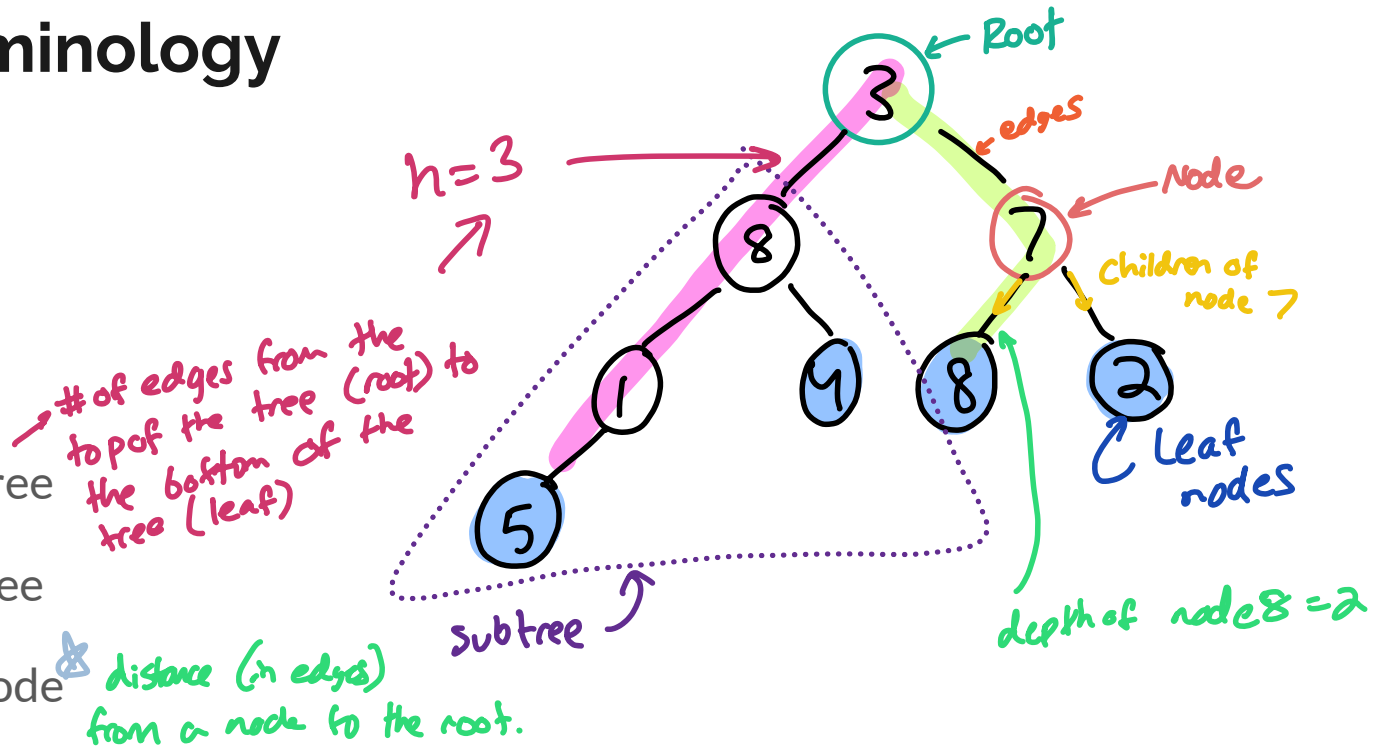
BST



"Invariants" dictate how values appear in the tree.

Tree Terminology

- Node
- Root
- Leaf
- Height of Tree
- Width of Tree
- Depth of Node
- Binary Tree



Other Notes on Trees

Binary

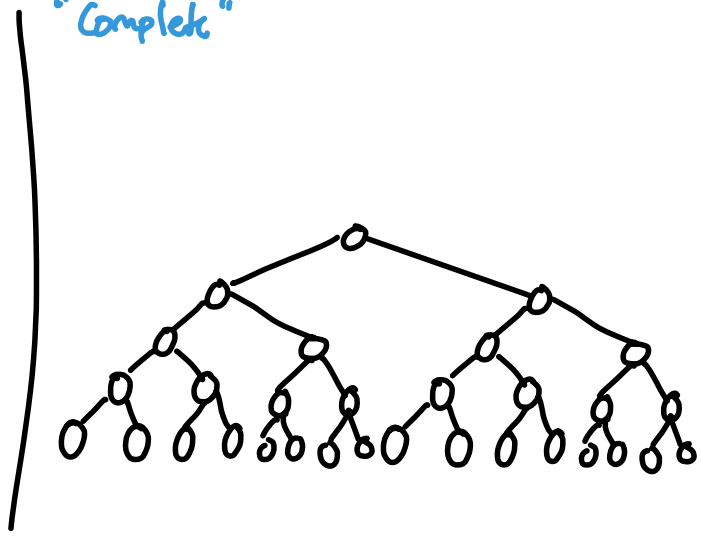
Given height H:

- Maximum Number of Nodes? $= 2^{(h+1)} - 1$
- Minimum Number of Nodes? $= h+1$

T=?

h=4

"Complete"



h=0

$2^{(0+1)} - 1$

$2^1 - 2 = 1$

h=1

$2^{(1+1)} - 1$

$4 - 1 = 3$

h=2

$2^{(2+1)} - 1$

$8 - 1 = 7$

...

Tree Traversals

```
void traverse(Tree t) {
  if(t == null) return;
    print(t.getValue();
    traverse(t.getLeft());
    traverse(t.getRight());
}
```

```
    traverse(t.getLeft());
    print(t.getValue();
    traverse(t.getRight());
```

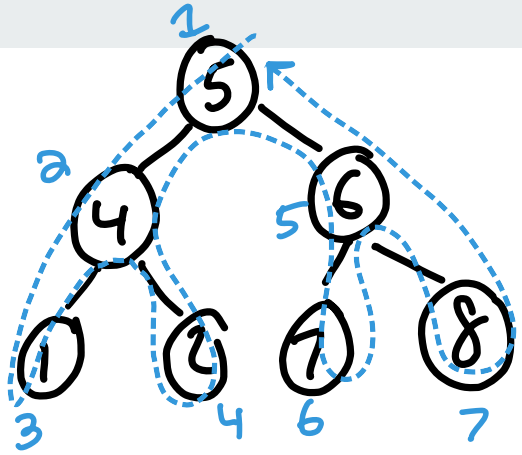
```
    traverse(t.getLeft());
    traverse(t.getRight());
    print(t.getValue();
```

Pre-order

In order

post-order

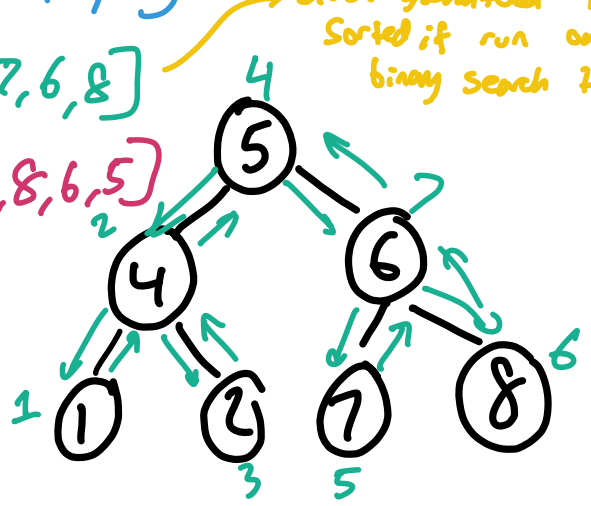
Tree Traversals

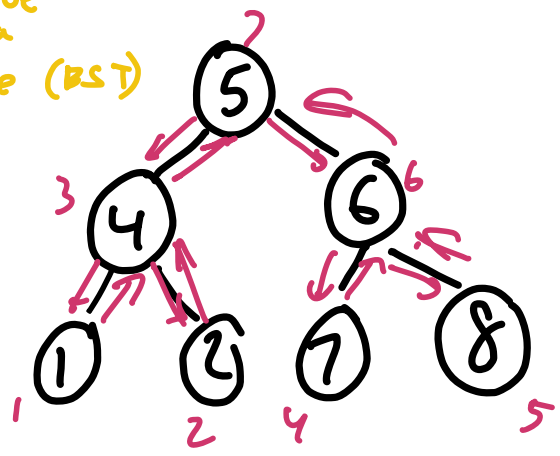


Find the...

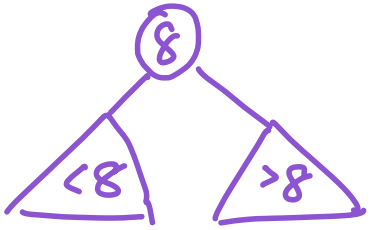
- Pre-Order Traversal: [5, 4, 1, 2, 6, 7, 8]
- In-Order Traversal: [1, 4, 2, 5, 7, 6, 8]
- Post-Order Traversal: [1, 2, 4, 7, 8, 6, 5]

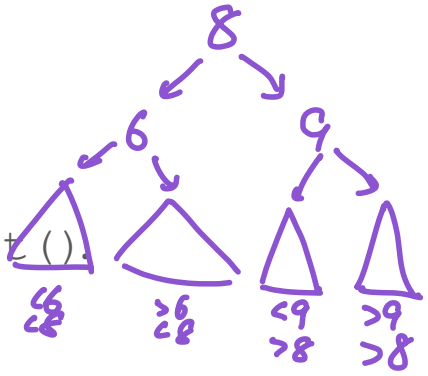
ONLY guaranteed to be Sorted if run on a binary search tree (BST)





Binary Search Trees

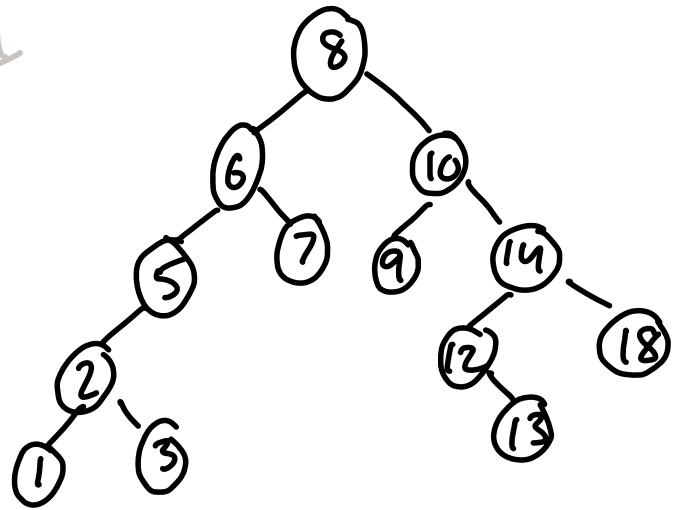




- Binary Trees where given ANY root r in the tree:
    - Every value in `r.getLeft()` < `r` < every value in `r.getRight()`

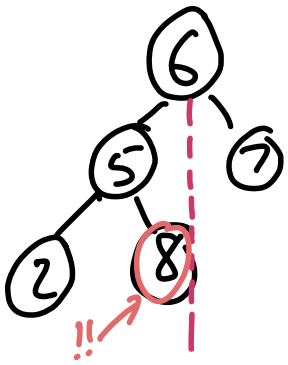
Is this a BST?

#1



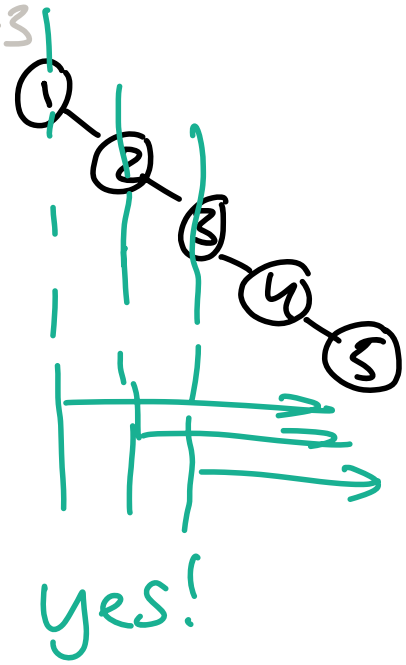
yes!

#2



no!

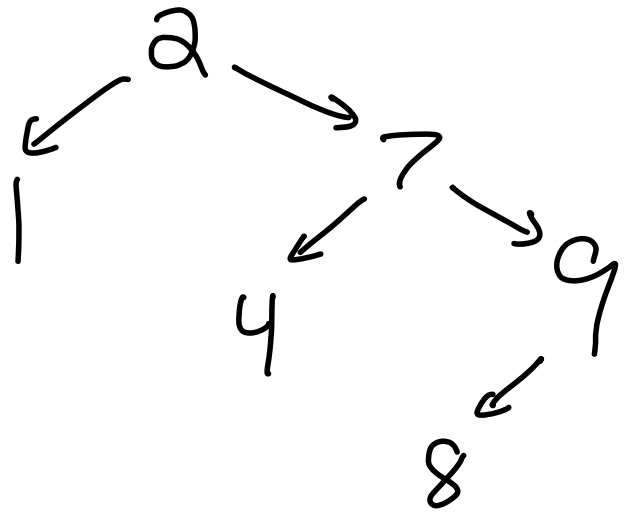
#3



yes!

Creating a BST

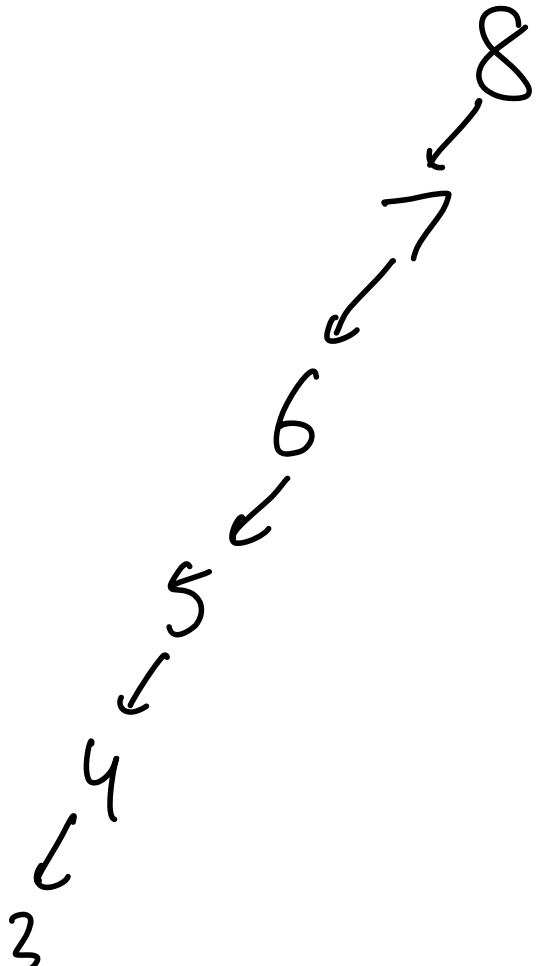
Inputs: [2, 7, 4, 1, 9, 8]



For any collection of inputs, there are many possible trees that can be made.

Creating a BST

Inputs: [8, 7, 6, 5, 4, 3, 2, 1]



2
1