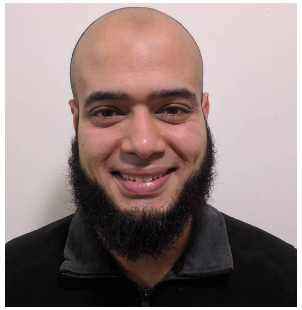
Data Structures

Level Order Traversal 1

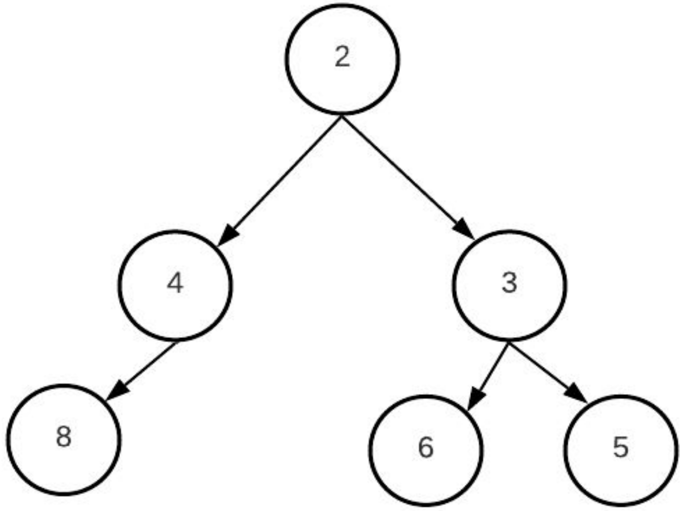
Mostafa S. Ibrahim
Teaching, Training and Coaching since more than a decade!

Artificial Intelligence & Computer Vision Researcher
PhD from Simon Fraser University - Canada
Bachelor / Msc from Cairo University - Egypt
Ex-(Software Engineer / ICPC World Finalist)



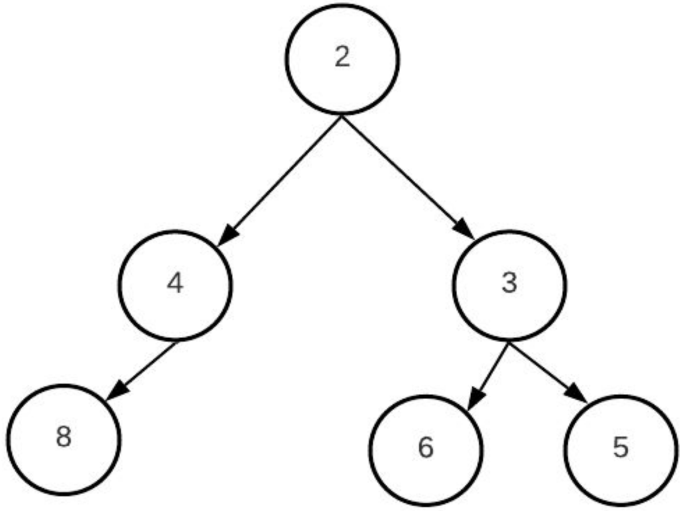
Level Order Traversal

- We learned 3 recursive traversal methods that each go as deep as possible
  - We call them depth first (go deeper)
  - Inorder here is: 8 4 2 6 3 5
- In level order traversal, we print the tree level by level
  - Level 0: 2
  - Level 1: 4 3
  - Level 2: 8 6 5
  - We call it: breadth first
- Depth vs Breadth



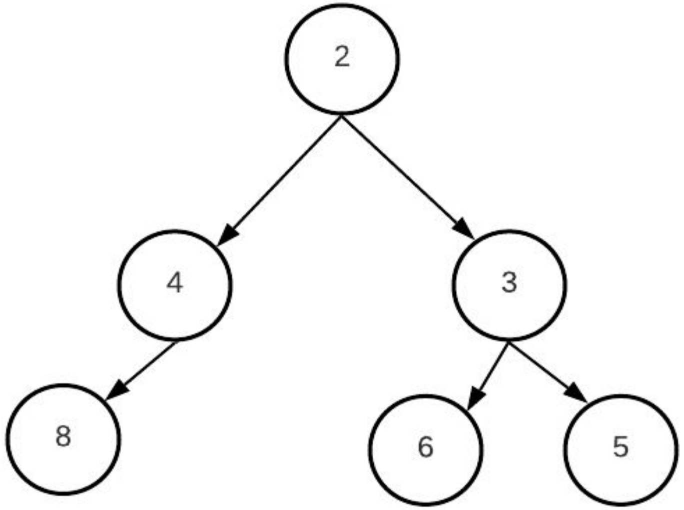
Traversing level by level

- Although we can use recursion to find the levels one by one, it will be very impractical
- One of the great applications of the **queue** is to iterate on a tree level by level
- Start a queue with the root.
- Pop the root node, then immediately push the root node's children - and then repeat this process with each child node!
- Can you finish this idea, and implement it by yourself?



Traversing level by level

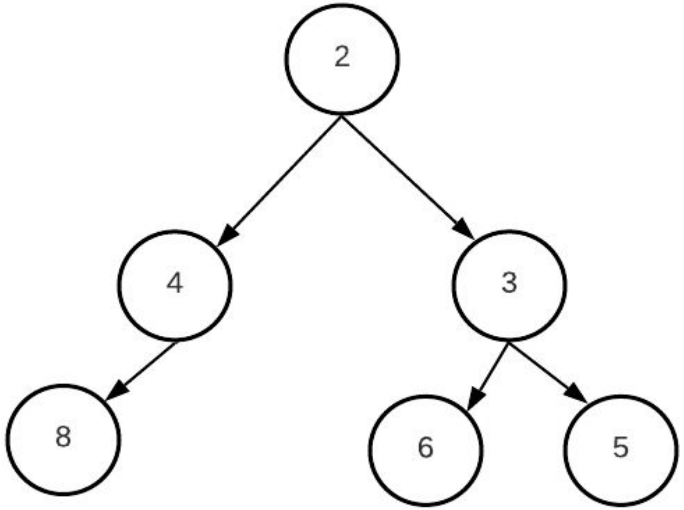
- Add the root node to the queue
- While not empty
  - Get node
  - Print it
  - Add its available children



2				

Traversing level by level

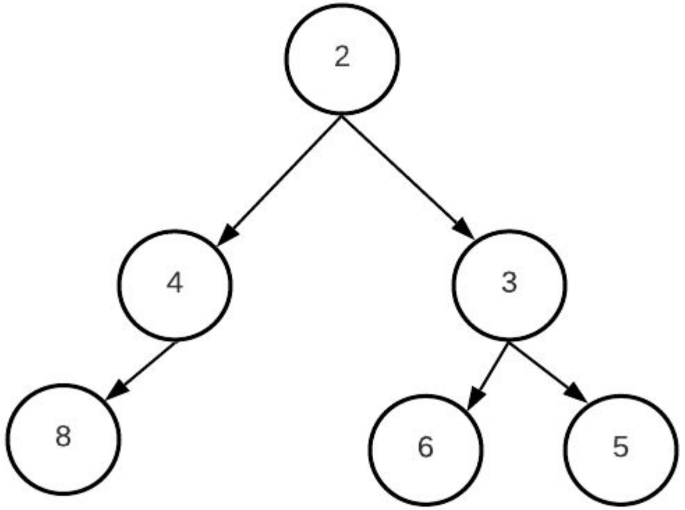
- Pop: Tree(2)
- Add children: 4, 3
- **Printed so far**: 2



4	3			

Traversing level by level

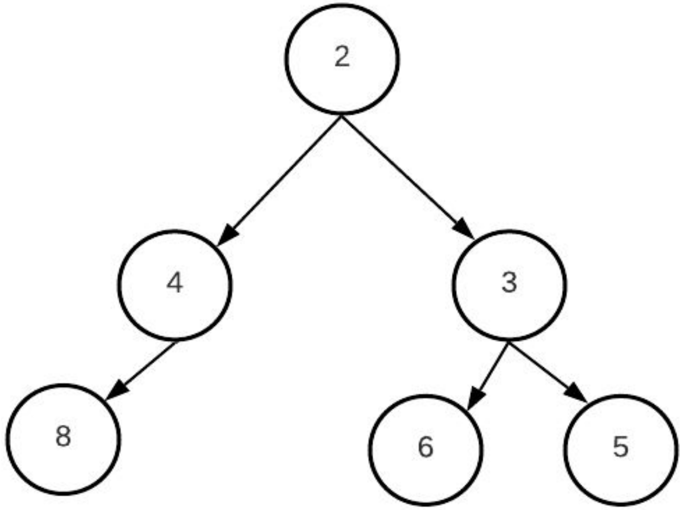
- Pop: Tree(4)
- Add children: 8
- **Printed so far**: 2 4



3	8			

Traversing level by level

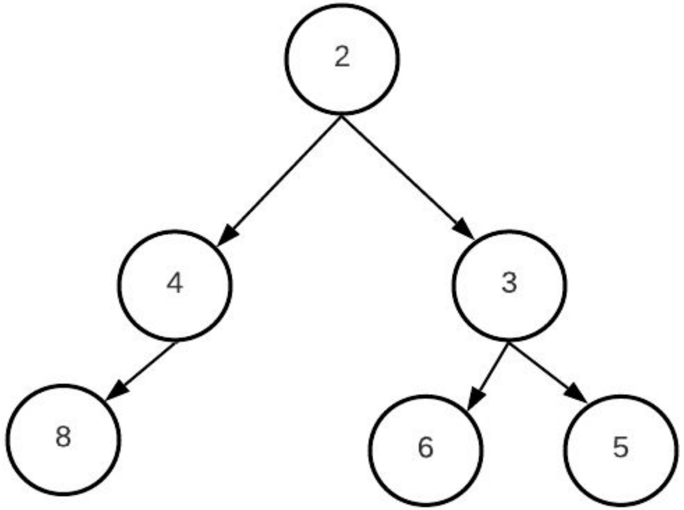
- Pop: Tree(3)
- Add children: 6, 5
- **Printed so far**: 2 4 3



8	6	5		

Traversing level by level

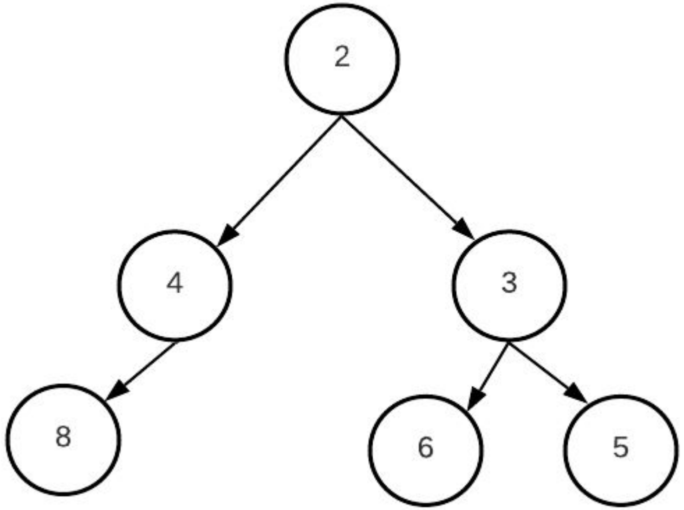
- Pop: Tree(8)
- Add children: None
- **Printed so far**: 2 4 3 8



6	5			

Traversing level by level

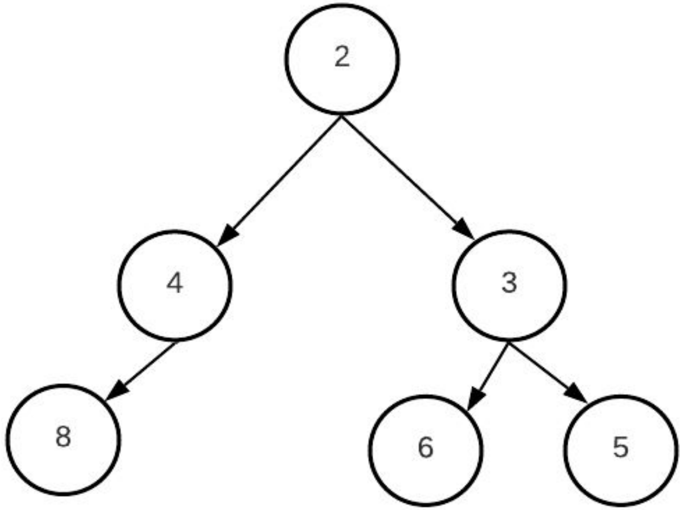
- Pop: Tree(6)
- Add children: None
- **Printed so far**: 2 4 3 8 6



5				

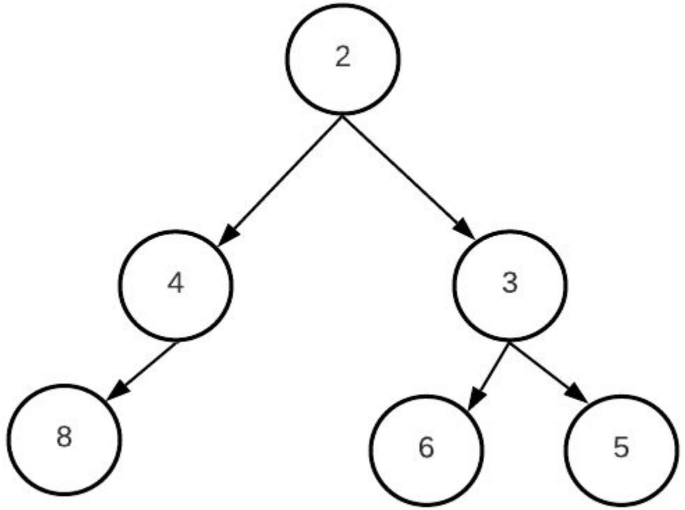
Traversing level by level

- Pop: Tree(5)
- Add children: None
- Empty: queue: stop
- **Printed so far**: 2 4 3 8 6 5



The queue content

- What is happening? We iterate through the nodes one by one, adding each node's child/children to the current queue
- The queue will be in 1 of 2 cases
  - Either all current nodes are at 1 specific level
  - OR it will contain nodes from 2 consecutive levels

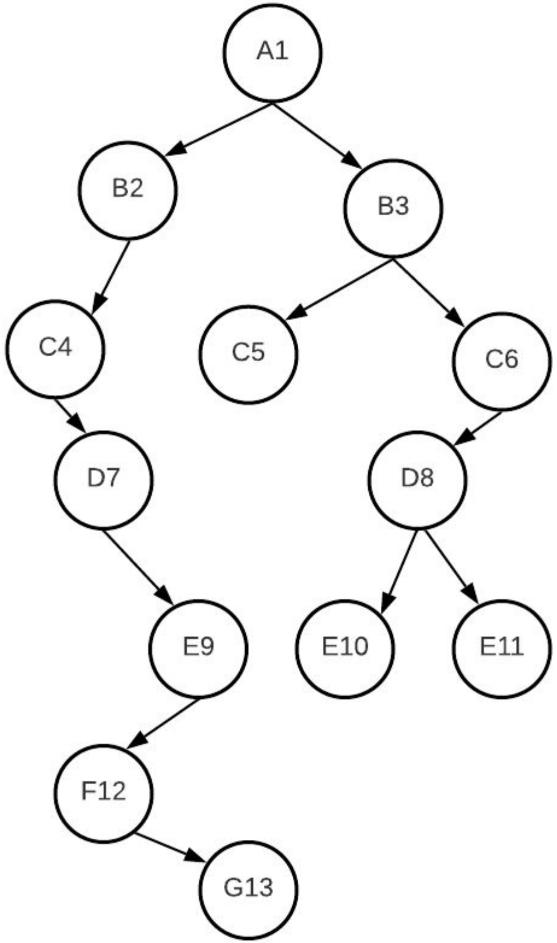


3	8			

8	6	5		

Let’s check the queue

● A1	: remove A1, add B2, B3
● B2, B3	: remove B2, add C4
● B3, C4	: remove B3, add C5, C6
● **C4, C5, C6**	: remove C4, add D7
● **C5, C6, D7**	: remove C5, add nothing
● C6, D7	: remove C6, add D8
● D7, D8	: remove D7, add E9
● D8, E9	: remove D8, add E10, E11
● E9, E10, E11	: remove E9, add F12
● E10, E11, F12	
● F12	
● G13	



“Acquire knowledge and impart it to the people.”

“Seek knowledge from the Cradle to the Grave.”