Data Structures

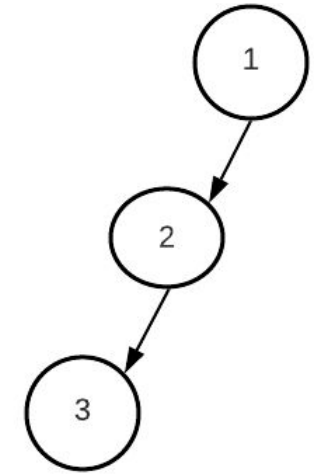
Binary Tree Generation

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

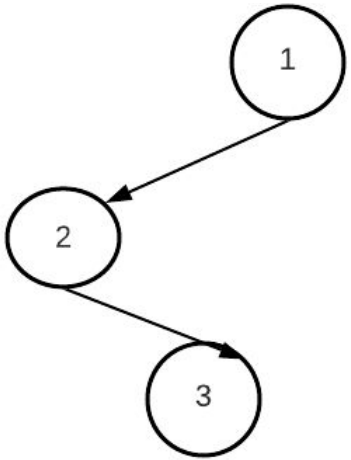
Artificial Intelligence & Computer Vision Researcher
PhD from Simon Fraser University - Canada
Bachelor / Msc from Cairo University - Egypt
Ex-(Software Engineer / ICPC World Finalist)

From a traversal to tree

- *Recall: there are many (un)labeled trees*
- Given a binary tree: there is a **unique** preorder, inorder and postorder
- But given **one** of the traversal orders, is there is a **unique** tree?
  - **The answer is no**
- Even worse: Given both preorder and postorder, is there is a unique tree?
  - **The answer is no**
  - On right, a counter example!
  - Different trees with same preorder & postorder!
  - Intuition: both styles hold similar info



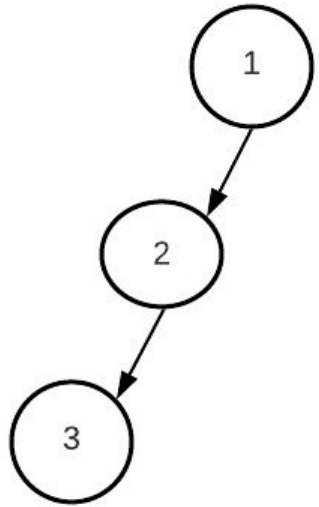
Preorder: 123
Postorder: 321



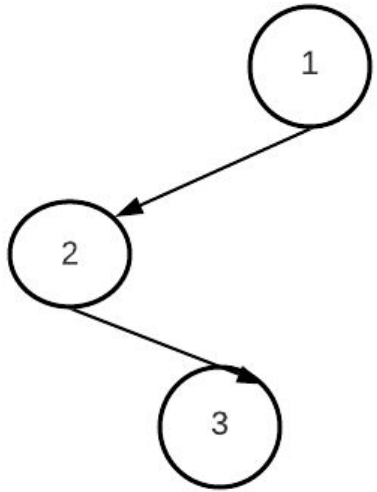
Preorder: 123
Postorder: 321

Given both inorder and preorder, can we?

- Given both **inorder** and preorder, can we generate a unique tree?
  - *Assume nodes values are unique*
- YES. Their information is complementary
  - Can u find 2 key observations?



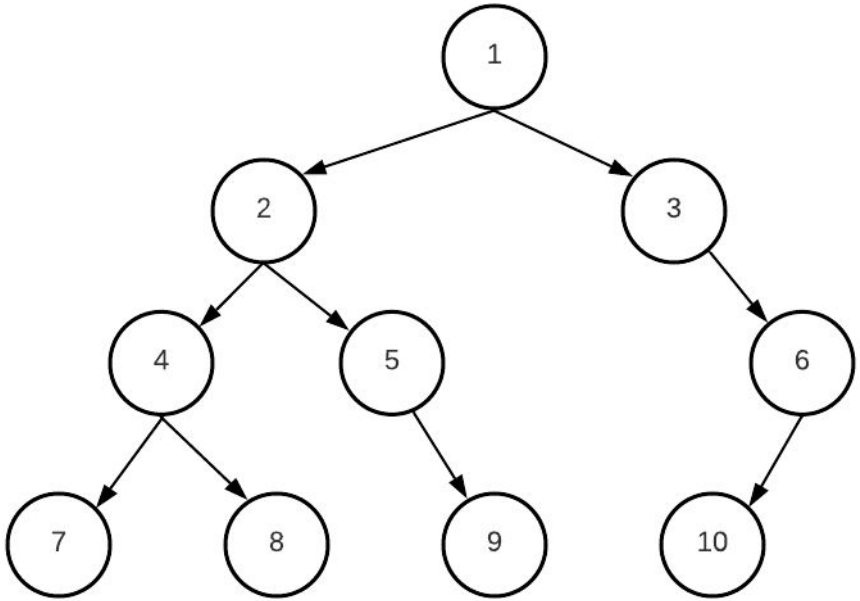
Preorder: 123
Inorder: 321



Preorder: 123
inrder: 231

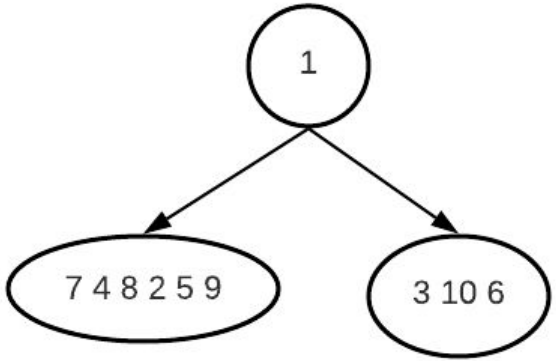
Let’s try an example

- Preorder: [1 2 4 7 8 5 9 3 6 10]
  - Observe: root is first value
- Inorder: [**7 4 8 2 5 9** 1 **3 10 6**]
  - Observe: 1 split in-order to
    - [**7 4 8 2 5 9**] = left subtree
    - [**3 10 6**] = right subtree
- This property holds recursively



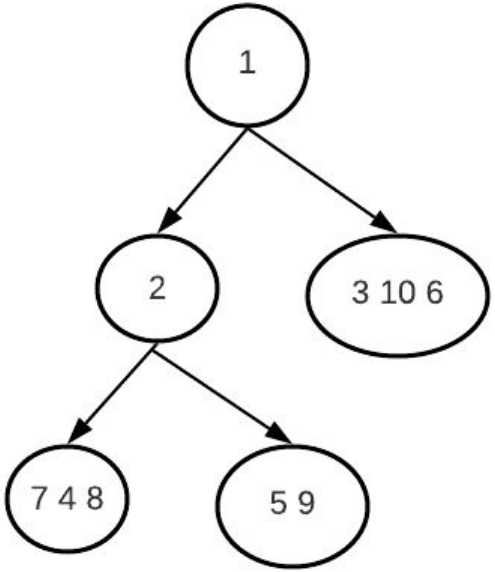
Pick next root

- Preorder: [2 4 7 8 5 9 3 6 10]
  - Preorder goes left first
  - Is 2 in the left sub-tree? No guarantee
- The next value (2) in pre-order is a 2nd root
- Where is 2? In subtree [7 4 8 2 5 9]
  - Again this splits the inorder to 2 parts (before/after)
  - **7 4 8** 2 **5 9**
  - [**7 4 8**] = left subtree
  - [**5 9**] = right subtree



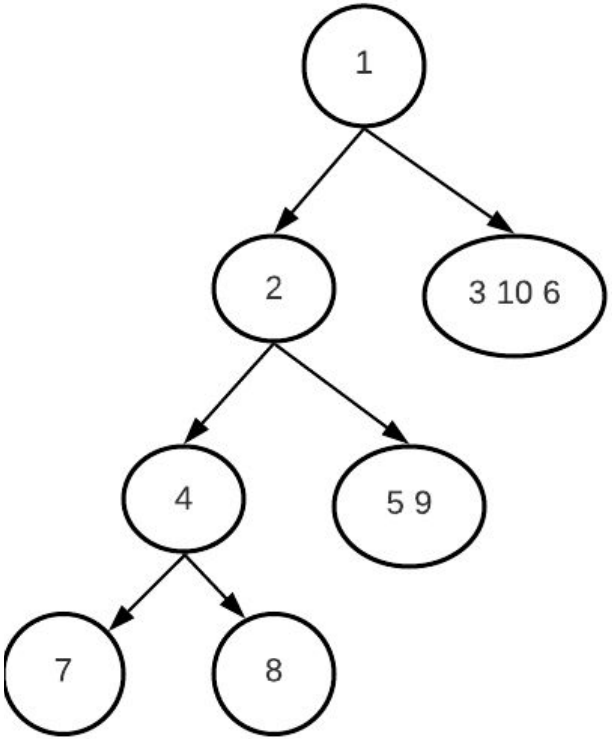
Pick next root

- Preorder: [4 7 8 5 9 3 6 10]
- Next root is 4
- Where is 4? In left subtree: [7, 4, 8]
  - [**7**] = left subtree
  - [**8**] = right subtree



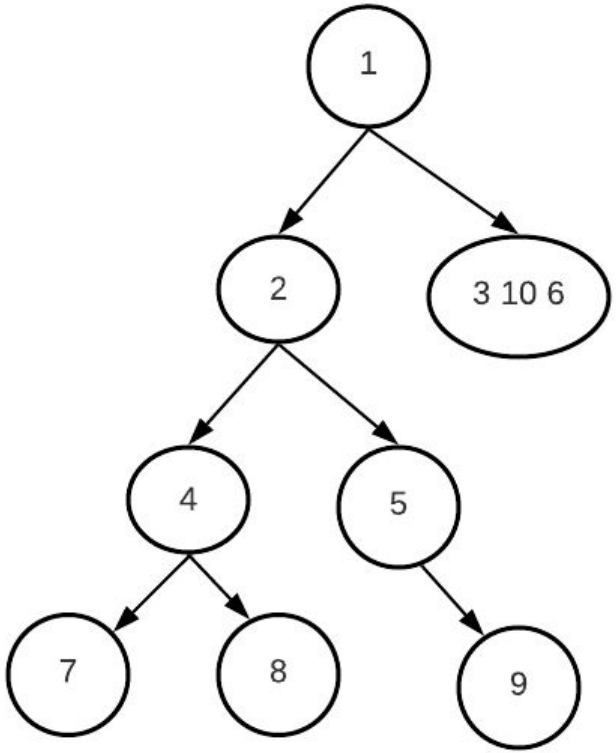
Pick next root

- Preorder: [4 7 8 5 9 3 6 10]
- Next root is 7 in subtree [7]
- Next root is 8 in subtree [8]
- Next root is 5 in subtree [5, 9]
  - [] = left subtree
  - [**9**] = right subtree



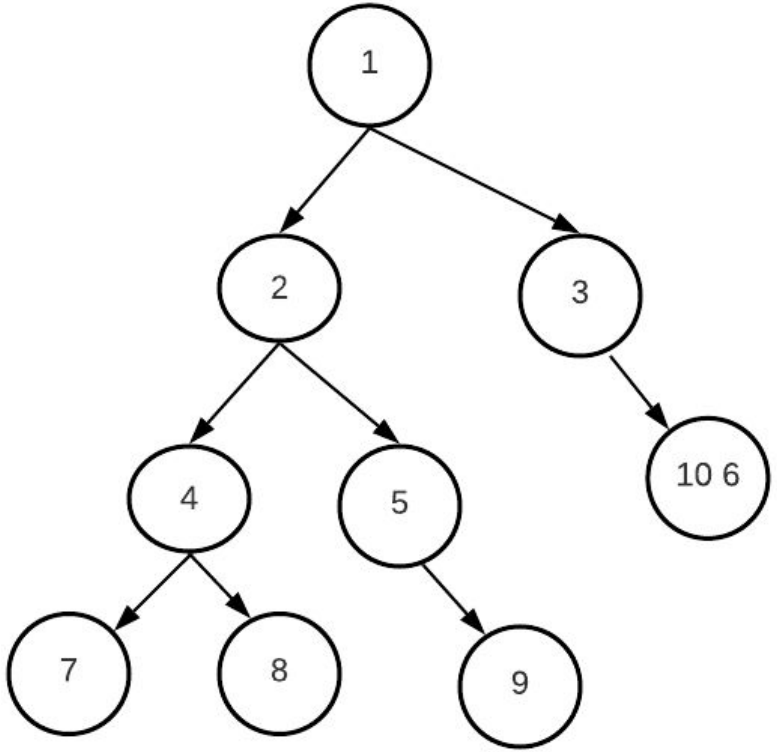
Pick next root

- Preorder: [9 3 6 10]
- Next root is 9 in subtree [9]
- Next root is 3 in subtree [3, 10, 6]
  - [] = left subtree
  - [**10, 6**] = right subtree



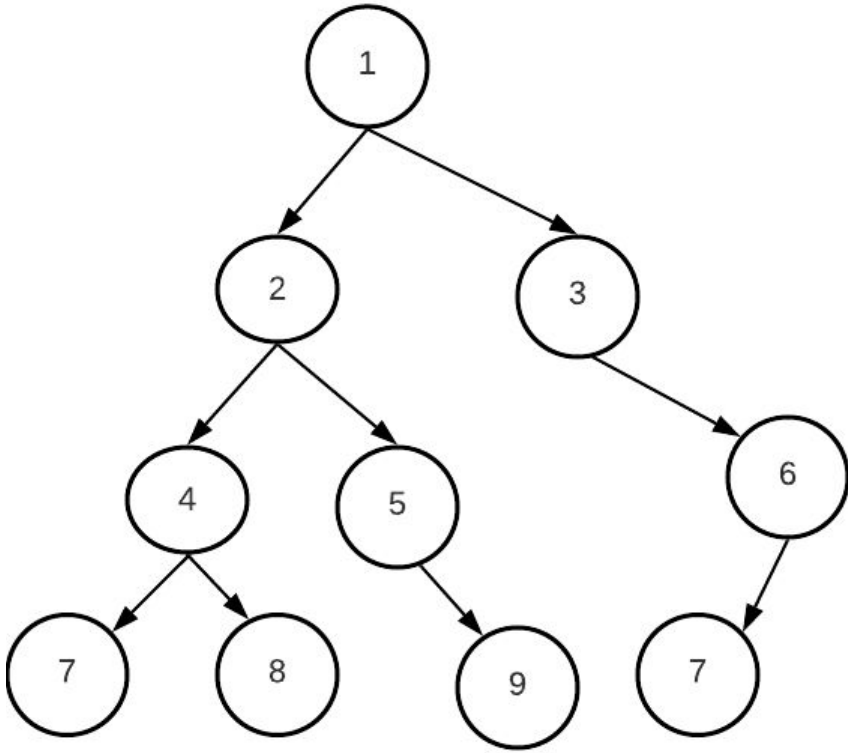
Pick next root

- Preorder: [6 10]
- Next root is 6 in subtree [10, 6]
  - [10] = left subtree
  - [] = right subtree



Pick next root

- As you see, by taking root by root from the preorder and deciding where is corresponding subtree and how to split it, we managed to build a unique tree
- The combined info of preorder and inorder allows that
- Similarly, the postorder & inorder are complementary



Take-Home message

- Given a binary tree: there is a **unique** preorder, inorder, postorder, **level order**
- 3 cases that **uniquely** define a tree: **inorder** + (preorder, postorder or level)
- Preorder+postorer, preorder+level-order, postorder+level-order **can't define** a unique **general** tree!
- However, it might be possible for the full binary tree case (0 or 2 children)
  - See homework
- Thinking Tip:
  - Some **general** algorithms can be applied differently/efficiently on a **constrained** problem
  - It is common mistake to assume special versions must follow the general version
  - Whenever you try to solve a problem, make use of its **special characteristics**!
    - General binary tree, full tree, perfect, complete, balanced, degenerate, etc

"Acquire knowledge and impart it to the people."

"Seek knowledge from the Cradle to the Grave."