

BST Mutation: Delete

Arnamoy Bhattacharyya

Warmup exercise: find node with min value in BST

delete() operation

- **Input:**

- BST rooted at a node called *root*
- a value *V* that we wish to delete from the tree (if it exists!)

- **Output:**

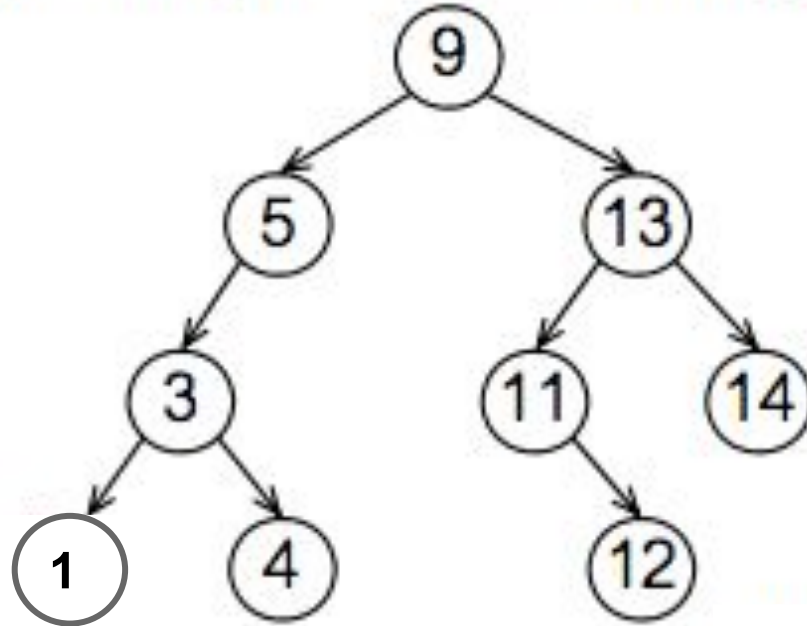
- if *V* found => return corresponding node from the tree.
- This node gets “disconnected” / ”extracted” / “removed” from the tree
- if *V* not found => return None

Deletion of data from BST rooted at 'node'?

- Locate the node to delete, by traversing the tree
- Possible scenarios
 - What to do if current is None?
 - What if data to delete is less than the data within the current node?
 - What if data to delete is more than the data within the current node? •
 - What if data to delete equals current's data?

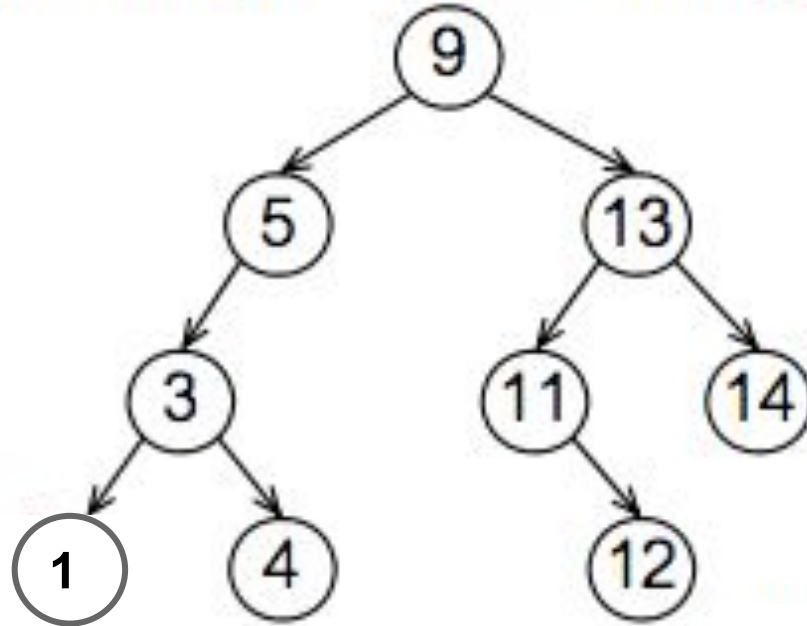
Data to delete == current_node.data

Delete 1?



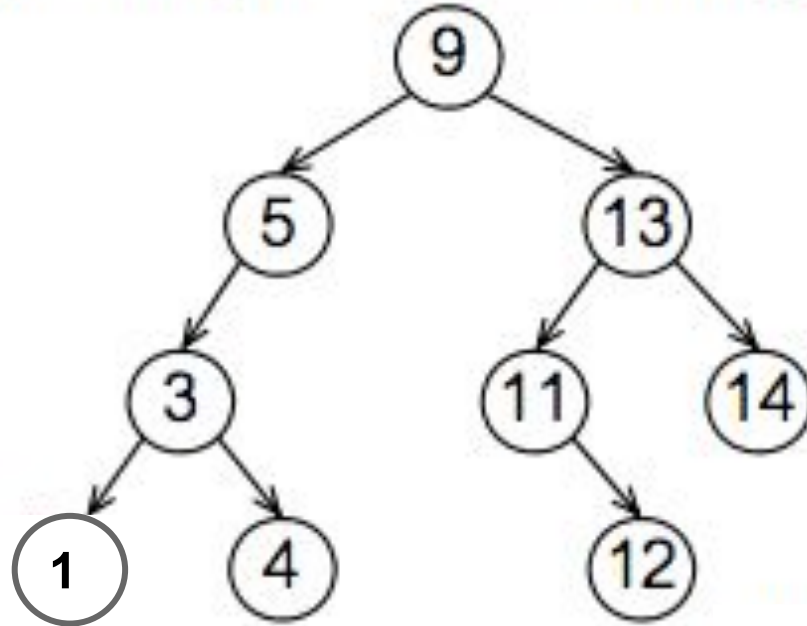
Data to delete == current_node.data

Delete 5?



Data to delete == current_node.data

Delete 9?



Delete three scenarios

1. Current has **no** child -- easiest (just remove)
2. Current has **one** child -- tad harder
3. Current had **two** children -- hardest

Implementation of delete: steps()

```
def delete(node: Union[BinaryTree, None],  
           data: object) -> Union[BinaryTree, None]:
```

- If current is **None** => not found, done!
- If data to delete is **less** than current's data => inspect left subtree
- If data to delete is **more** than current's data => inspect right subtree
- If data to delete **equals** current's data
 - Case a) **No children** => easy, just remove the node
 - Case b) **One child** => easy, just connect current's child to current's parent
 - Case c) **Two children** => clear current's value, pick a replacement value from a descendant under it, and delete that descendant node
 - Max from left subtree, or min from right subtree

delete() implementation

Onto Pycharm

Term test topics

1. Tracing recursive functions (avoid redundant tracing)
2. Recursion on nested lists
3. Problems on Trees
4. Problems on Binary Trees
5. Definition and terminology of BST and traversals (in, pre, post)
6. BST insert/ delete NOT included