

# Practice Interview

## Objective

*\*The partner assignment aims to provide participants with the opportunity to practice coding in an interview context. You will analyze your partner's Assignment 1. Moreover, code reviews are common practice in a software development team. This assignment should give you a taste of the code review process.\**

## Group Size

Each group should have 2 people. You will be assigned a partner

## Part 1:

You and your partner must share each other's Assignment 1 submission.

## Part 2:

Create a Jupyter Notebook, create 6 of the following headings, and complete the following for your partner's assignment 1:

- Paraphrase the problem in your own words.

```
In [ ]: The problem asks us to examine a binary tree and determine if it contains duplicate
```

- Create 1 new example that demonstrates you understand the problem.  
Trace/walkthrough 1 example that your partner made and explain it.

```
In [ ]: 4
         / \
        3  6
       / \  \
      5  3  7
In this tree:

The root node has value 4.
Its left child has value 3, and its right child has value 6.
The left child of 3 is 5, and the right child of 3 is also 3 (duplicate).
The right child of 6 is 7.
Output:
3
```

```

      1
     /\
    10 2
   /\ /\
  3 10 12 12

```

Output:

10

Explanation:

Start at the root (1): Look at the root node with the value 1. Add 1 to a list of v

Go to the left child (10): Move to the left child of the root, which has the value

Go to the left child of 10 (3): Now look at the left child of 10, which has the val

Check the right child of 10 (10): Go back to the parent node 10 and now check its r

Found a duplicate (10): Since 10 is already in the list of seen values, this is the

- Copy the solution your partner wrote.

```

In [ ]: from collections import deque
        # Definition for a binary tree node.
        class TreeNode(object):
            def __init__(self, val=0, left=None, right=None):
                self.val = val
                self.left = left
                self.right = right

        def is_duplicate(root: TreeNode) -> int:
            # Handle edge case of empty tree
            if root is None:
                return -1

            # Initialize BFS
            queue = deque([root])
            visited = set()

            # Traverse the tree using BFS
            while queue:
                current_node = queue.popleft()

                # Check for duplicate
                if current_node.val in visited:
                    return current_node.val
                visited.add(current_node.val)

                # Add children to the queue
                if current_node.left:
                    queue.append(current_node.left)
                if current_node.right:
                    queue.append(current_node.right)

```

```
# If no duplicates found return -1
return -1
```

- Explain why their solution works in your own words.

In [ ]: The partner's solution is correct because it effectively uses BFS to traverse the tree, tracks seen values with a set, and prioritizes finding the duplicate closest to the root. The algorithm is efficient and handles edge cases properly.

- Explain the problem's time and space complexity in your own words.

In [ ]: The time complexity of the solution is  $O(n)$ , where  $n$  is the number of nodes in the tree. The space complexity is also  $O(n)$  due to the queue used for BFS and the visited set.

- Critique your partner's solution, including explanation, and if there is anything that should be adjusted.

In [ ]: The partner's solution is efficient and well-suited for the problem, using Breadth-First Search (BFS). However, there are some areas where the solution could be improved. Adding more detailed comments and handling edge cases more explicitly would be beneficial.

## Part 3:

Please write a 200 word reflection documenting your process from assignment 1, and your presentation and review experience with your partner at the bottom of the Jupyter Notebook under a new heading "Reflection." Again, export this Notebook as pdf.

## Reflection

In [ ]: Reflection  
The process of completing Assignment 1 required a thorough understanding of binary trees and their traversal algorithms. During the presentation and review process with my partner, I appreciated the collaborative learning experience and the opportunity to discuss different approaches to solving the problem.



## Evaluation Criteria

We are looking for the similar points as Assignment 1

- Problem is accurately stated
- New example is correct and easily understandable
- Correctness, time, and space complexity of the coding solution

- Clarity in explaining why the solution works, its time and space complexity
- Quality of critique of your partner's assignment, if necessary

## Submission Information

 **Please review our Assignment Submission Guide**  for detailed instructions on how to format, branch, and submit your work. Following these guidelines is crucial for your submissions to be evaluated correctly.

### Submission Parameters:

- Submission Due Date: HH:MM AM/PM - DD/MM/YYYY
- The branch name for your repo should be: `assignment-2`
- What to submit for this assignment:
  - This Jupyter Notebook (assignment\_2.ipynb) should be populated and should be the only change in your pull request.
- What the pull request link should look like for this assignment:  
`https://github.com/<your_github_username>/algorithms_and_data_structures/`
  - Open a private window in your browser. Copy and paste the link to your pull request into the address bar. Make sure you can see your pull request properly. This helps the technical facilitator and learning support staff review your submission easily.

#### Checklist:

- ☐ Created a branch with the correct naming convention.
- ☐ Ensured that the repository is public.
- ☐ Reviewed the PR description guidelines and adhered to them.
- ☐ Verify that the link is accessible in a private browser window.

If you encounter any difficulties or have questions, please don't hesitate to reach out to our team via our Slack at `#cohort-3-help`. Our Technical Facilitators and Learning Support staff are here to help you navigate any challenges.

