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.

The problem is that we need to return a list of numbers that are missing from the range of 0 to n where n is a number. If there are no missing numbers between 0 and n, print -1.

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

```
In [ ]:   # Example 1
          # Input: [1, 5, 10, 15]
          # Missing Numbers: [2, 3, 4, 6, 7, 8, 9, 11, 12, 13, 14, 15 ]
          # Output: [2, 3, 4, 6, 7, 8, 9, 11, 12, 13, 14, 15]
```

```
In [ ]:   # Example 2:
          # Input [0,4]
          # Missing Numbers: [1,2,3]
          # Output: [1,2,3]
```

In example 2, the list of numbers was 0 and 4, where 4 is the number n. The range of missing numbers and output was therefore 1, 2 and 3. If the input was 0, 1, 2, 3 and 4, then the output would've been -1 because there were no missing numbers.

- Copy the solution your partner wrote.

```
In [23]:  from typing import List

          def missing_num(nums: List[int]) -> List[int]:
              n = len(nums)
              missing = []

              for i in range(n + 1):
                  if i not in nums:
                      missing.append(i)

              if len(missing) == 0:
                  return [-1]
              else:
                  return missing
```

```
In [34]:  missing_num([0, 1, 10])
```

```
Out[34]:  [2, 3]
```

- Explain why their solution works in your own words.

```
In [ ]:   # They created an empty list to store the missing numbers.
          # They created a loop that goes through the range of numbers and if i is not found in nums, then they append missing numbers to "missing".
          # If the length of missing numbers is 0, -1 is returned because that means there are no missing numbers.
```

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

```
In [ ]:   # The problem's time complexity is O(n²). The space complexity is O(n).
```

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

```
In [ ]:   # Instead of using a list of numbers to search against, a set would also work as there are no duplicate numbers involved and it would reduce the time complexity.
          # For a larger range of numbers, this would be more relevant. In addition, making n equal to the length of nums works for some ranges but doesn't work for others
          # e.g. when nums is equal to [0, 1, 10], the length of nums is 3 so n becomes equal to 3; when the loop occurs, it will go through a range of 4 numbers, which isn't correct.
```

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 [ ]:   # For my problem, I went over it, the examples and the expected inputs and outputs.
          # I created examples that could be used to test the code and produce the different potential outcomes.
          # I selected a tree traversal method and created a basic solution based on the traversal method.
          # I then iterated on the solution ways to retain and store the seen values and duplicates.
          # I then added a helper function to traverse the tree and help produce the seen values and duplicates.

          # To review my partner's code, I first went over the problem they were trying to solve and reviewed the examples.
          # I then created my own example to ensure I understood what the inputs and outputs should look like.
          # I then ran the code to determine if the code was producing the expected output,
          # which it was for certain use cases and wasn't for others.
          # I documented  the parts that were working well and identified areas of improvement in terms of efficiency and efficacy.
          # To identify the areas of improvement in terms of efficacy,
          # I debugged the code to determine why it wasn't producing the expected output for all use cases.
```

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/pull/<pr_id>`
    - 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.