<> Code  Pull requests 1  Actions  Projects  Wiki  Security  Insights ⚙

[algorithms_and_data_structures](#) / [02_activities](#) / [assignments](#) / **assignment_2.ipynb** 📋  ⋯

Tarun Vashisht  assignment-2                    81da41f · 2 minutes ago  🕘

294 lines (294 loc) · 9.45 KB

Preview    Code    Blame                                    Raw 📋 ⬇  ✏ ▾

# 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 [ ]:
```
# Your answer here

# User has a list of n integers that are supposed to be within the range from 0 t
#  Some numbers in this range may be missing from the list,
#  and some numbers may repeat as well
#  User's task is to identify and return the missing numbers from this range.
#  If no numbers are missing, it should return -1
```

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

In [ ]:
```
# Your answer here
#my example:
# input[1,5,7,3]--> output [2,4]


# test_list3 = [8, 2, 5, 7, 0, 1, 10]
# output = [3,4,6,9]

# it is a list of random numbers with maximum value of 10. we intend to find out
# As per my partner's solution, he first find out the maximum number and then cal
# then he compared if the full range sum is equal to provided number list. If yes
# to find out the missing value.
```

- Copy the solution your partner wrote.

In [2]:
```
# Your answer here
def missing_num(nums: list[int]) -> int:
    if not nums:
```

```
    if not nums:
        return None

    n = max(nums)
    size = len(nums)
    expected_sum = n * (n + 1) // 2  # Calculate sum of [0, n]
    actual_sum = sum(nums)

    if expected_sum == actual_sum and size==n+1:
        return -1

    # Count occurrences of each number
    seen = set(nums)
    missing_numbers = []
    for i in range(n + 1):
        if i not in seen:
            missing_numbers.append(i)

    return missing_numbers
```

- Explain why their solution works in your own words.

In [ ]:
```
# Your answer here
# Solution works as it uses the sum of the given numbers/range and total sum of t
# no missing value. If no missing value, then it returns -1. If there are some mi
# loop to the range to the max of the set and checks if number is present or not.
```

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

In [ ]:
```
# since here for loop has been used then time and space complexity is O(1)
```

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

In [15]:
```
# Your answer here
# Solution is missing one requirement of having duplicate in number set. If there
# list is same as max+1 then solution fails.
# Eg. below values when passed to solution will give output -1, which is incorrec
# test_list3 = [3, 1, 1, 1]
# instead of using FOR loop there would have been a better solution of using SET
# it would have been faster and with lesser piece of code to achieve the same fun
# missing_numbers = sorted(list(full_set - num_set))
```

In [16]:
```
test_list3 = [3, 1, 1, 1]
print("Custom Test 3: ", missing_num(test_list3))
```

Custom Test 3:  -1

## 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 [18]:
```
# Your answer here
# The problem statement for my assignment-1 was clear and concise. it helped me t
```

```
# reading and proof reading.
# Solution based on the requirements. I learnt that we have to be vigilant when w
# with my problem statement, Thorough with Binomial trees.
# It was a good experience of reading my partner's solution and learnt that we ca
# by using a formulae.
```

# 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.