

Show Me The Data Structures

REVIEW

CODE REVIEW 12

HISTORY

Requires Changes

2 specifications require changes

- Hello great developer 😊 it's really great to work!
- I swear you will be a good developer 🙌🙌🙌
- You've done a nice job 📁+1: I must say, I'm impressed 🙌👏🎉 with your effort.
- You have made a very good User Interface and the code quality was great.?? ??
- You are close to complete the project as the required changes are very small.
- Awesome work on the project, everything looks great! I have added some tips and resources to your code review if you have time you can take a look! Hope you will find them helpful! ??
- Your hard work has paid off! Keep up the good work as you continue your Nanodegree journey! Safe journey and bon voyage! 😊
- Don't get upset or disappointed, you did a great job which deserves a big compliment, think that those changes are a great opportunity to learn more and perfect your skills.
- Please see the code review, it will help you to complete the project
- Also, look at these resources to learn more about **Python**:

- [Resource 1](#)
- [Resource 2](#)

Stay 🙌!

Code produces the correct solution to the question. There are also no runtime or compile time errors.

- Awesome!! I can see that your **output** is correct.
- For Example, in **problem 5**, I can see that you are calculating the **hash** correctly.
- Well done 🙌🙌

```
import json
import hashlib
import datetime

class DoublyNode:
    def __init__(self, value=None):
        self.value = value
        self.prev = None
        self.next = None

    def get_value(self):
        return self.value

    def set_value(self, value):
        self.value = value

    def __repr__(self):
        return str(self.value)

class DoublyLinkedList:
    def __init__(self):
        self.head = None
        self.tail = None
        self.length = 0

##### - Test Case 1 - #####
0
1
{"index": 0, "time_stamp": "2021-10-16T14:24:10.525945UTC", "data": "Record 1", "hash": "6585d2275383d692c67b3e7376345658371f3b65ab428c98b62f9865f9673866", "prev_hash": 0}
{"index": 0, "time_stamp": "2021-10-16T14:24:10.525945UTC", "data": "Record 1", "hash": "6585d2275383d692c67b3e7376345658371f3b65ab428c98b62f9865f9673866", "prev_hash": 0}
2
{"index": 0, "time_stamp": "2021-10-16T14:24:10.525945UTC", "data": "Record 1", "hash": "6585d2275383d692c67b3e7376345658371f3b65ab428c98b62f9865f9673866", "prev_hash": 0}
{"index": 1, "time_stamp": "2021-10-16T14:24:10.526094UTC", "data": "Record 2", "hash": "9deb8954af9956842a66f53d2e4fe5589f6c26eee8bbb49da342b0cda83ab6fb", "prev_hash": "6585d2275383d692c67b3e7376345658371f3b65ab428c98b62f9865f9673866"}
3
{"index": 0, "time_stamp": "2021-10-16T14:24:10.525945UTC", "data": "Record 1", "hash": "6585d2275383d692c67b3e7376345658371f3b65ab428c98b62f9865f9673866", "prev_hash": 0}
{"index": 2, "time_stamp": "2021-10-16T14:24:10.526160UTC", "data": "Record 3", "hash": "e87917a8bb62d38782dd765c83846886a1f34f48e685d73aa6768d72e0e15932", "prev_hash": "9deb8954af9956842a66f53d2e4fe5589f6c26eee8bbb49da342b0cda83ab6fb"}
```

Code is neat and easy-to-read. Variables, functions, and methods have straightforward names. There is enough spacing that code is easily readable.

- Awesome!! Your code is neat and **easy-to-read**. Variables, functions, and methods have **straightforward names**.
- For example: `calc_hash(self)` function in **problem 5** has a **straightforward** name 🙌🙌.
- Browse this [link](#) to learn more about how to make your code cleaner.

Code solution is not unnecessarily complex—it accomplishes the task at hand without extra iterating, algorithms, data structures, et cetera.

Testing

At least three test inputs and outputs are provided. There are at least two that test for edge cases, like null or empty inputs, or very large numbers.

- You made great work so far on your code and **test cases**.

- All problems contain all the required **test cases** 🙌🙌
- Browse this [link](#) to learn more about the **edge cases**.

Explanation

There is a clear and accurate statement of efficiency in time and space. There is an explanation that specifically mentions parts of the code that contribute to the overall efficiency.

- You made great work so far on your **analysis** and **explanation**.
- I can see that you correctly added **time complexity** to all your **explanation files** except **problem 3** you need to update the **time complexity** there.
- **Time complexity** for **problem3** should be `O(NLogN)`
- Please look at **the code review section** for more details.

Explanation contains some discussion of design choices made in the code. Some examples include the choice of algorithm and data structure.

Awesome!!! Your **explanation analysis** looks good and contains a discussion about design choices made in the code 🙌🙌

Explanation is written with proper English. Wording is clear and easy to understand. It's okay to make a couple mistakes, but thoughts should be clearly expressed overall.

 RESUBMIT

 DOWNLOAD PROJECT





Best practices for your project resubmission

Ben shares 5 helpful tips to get you through revising and resubmitting your project.

[🕒 Watch Video \(3:01\)](#)

RETURN TO PATH

Rate this project

START