Homework 2 Self-Evaluation

5/5 Points

1/25/2024

Attempt 1

Review Feedback
1/23/2024

Attempt 1 Score:
5/5

View feedback

Anonymous grading: **no**

Unlimited Attempts Allowed
1/22/2024 to 1/27/2024

Details

Please answer the following questions about the code you submitted to the **first** submission deadline (i.e., not the resubmission, and not your work in progress for the resubmission). If you overwrote it locally, you can redownload your submission from Canvas.

For each question, answer either with the line number (or a range of line numbers) that is relevant to the question, or with "no" if your code does not do what the question is asking about. *Your answer to a question will get 1 point if your answer accurately answers the question and you did not answer "no"; your answer will get a 0 in all other cases.*

Make sure to **double-check your answer and line numbers** to make sure they are correct and also that they are referencing the correct version of the file (see above). To ensure consistency in grading across all students, **your line numbers MUST correspond to the relevant lines of code, otherwise the question will get a 0** (even if the relevant code lies elsewhere).

---

1. Do you have a test that checks that, when you push two or more elements to one of your `ListStack`s, they come out in LIFO order? (You need a test with at least two pushes and two pops). Answer with the line where we can find this test.
2. Does your `ListStack.pop` method call `error` if the user attempts to pop from an empty stack? Answer with the line in your `pop` method where it does that.
3. Does your `ListQueue' class perform both enqueue and dequeue with O(1) efficiency by maintaining a tail arrow field? If so, answer with the line in your `enqueue` method where you update your tail pointer. (If you do it in multiple places, any of them is fine.) If your `enqueue` method loops or recurses through the list to perform an enqueue, answer with "no".
4. Did you test your `fill_playlist` function with at least two different concrete implementations of the QUEUE interface? If so, answer with the two lines at which we can find two of these tests.
5. *For this last question, answer with a short paragraph. We will not be looking for a specific correct answer, but rather for thoughtfulness and reflection.*

   Stacks and queues offer strict LIFO and FIFO order, respectively. Give a concrete, real-world

example where you'd want to keep track of a series of tasks that need to be done, but where neither strict LIFO or strict FIFO would be appropriate. Explain why it would not be appropriate, and describe an order that would be.

View Rubric

Select Grader

Ethan Pineda (TA)

Self-Eval 22-23

Criteria	Ratings		Points
Q1 view longer description	**1 pts** **Got it**	**0 pts** **Missing/Incorrect**	1 / 1 pts
Q2 view longer description	**1 pts** **Got it**	**0 pts** **Missing/Incorrect**	1 / 1 pts
Q3 view longer description	**1 pts** **Got it**	**0 pts** **Missing/Incorrect**	1 / 1 pts
Q4 view longer description	**1 pts** **Got it**	**0 pts** **Missing/Incorrect**	1 / 1 pts
Q5 view longer description	**1 pts** **Got it**	**0 pts** **Missing/Incorrect**	1 / 1 pts
			Total points: 5

1. 80
2. 57-58
3. 118
4. 188, 193
5. In a setting where tasks can be ordered by priority, eg, a company addressing user complaints, neither LIFO nor FIFO works. With LIFO, older complaints get perpetually ignored. With FIFO, more urgent issues may be deprioritized in favor of more recent ones. These tasks must be sorted by priority before being addressed.