Homework 3

Note: You may collaborate on the remainder of the assignment. If you do collaborate on the assignment, list your collaborators. All duplicate assignments without collaborators listed will be flagged for plagarism. Additionally, presenting others work – including a Chegg expert answer – as your own work *is* a violation of the academic dishonesty policy.

This homework assignment is designed to prepare you for the in class quiz.

1 Testing Exceptions

For the `MyArrayList.java` class in the homework zip folder, write a test for each of the following in a new Java file `MyArrayListTest`:

(a) Set Up a test fixture for `MyArrayListTest` with and `@BeforeEach` and `@AfterEach`. In comments above these test fixture, explain why you created this test fixture.

(b) For each test case provide the following:

- Write a test for `ensureCapacity` to make sure that it throws an `IllegalArgumentException` if the value passed is negative.
- Write a test for `get` to make sure it throws an `ArrayIndexOutOfBoundsException` if the value passed is negative.
- Write a test for `get` to make sure it throws an `ArrayIndexOutOfBoundsException` if the value passed is larger than the array.
- Write a test for `set` to make sure it throws an `ArrayIndexOutOfBoundsException` if the value passed is negative.
- Write a test for `set` to make sure it throws an `ArrayIndexOutOfBoundsException` if the value passed is larger than the array.
- Write a test for `set` to make sure it throws an `NullPointerException` if the value passed is null.
- Write a test for `remove` to make sure it throws an `IllegalStateException` if list is already empty.

2 Data-Driven ArrayList

(a) Write a data-driven, parameterized test for each of the following and put it in `DataDrivenMyArrayList.java`:

- Test `add(index)` for multiple inputs (at least 3) within one data-driven test. The test should ensure the appropriate element is added to the end of the array.
- Test `add(index, element)` for multiple inputs (at least 3) within one data-driven test. The test should ensure the appropriate element is added to the specified index in the array.
- Test `remove(index)` for multiple inputs (at least 3) within one data-driven test. The test should ensure that the appropriate element is removed from the array.
- Test `set(index, element)` for multiple inputs (at least 3) within one data-driven test. The test should check that the value and the index is updated to the element.
- Test `get(index)` for multiple inputs (at least 3) within one data-driven test. The test should check that the appropraite element is returned.
- Test `ensureCapacity` for multiple inputs (at least 3) within one data-driven test. The test should check that the ArrayList is actually sized correctly.
- Test `trimToSize` for multiple inputs (at least 3) within one data-driven test. The test should check that the ArrayList is actually sized correctly.
- Test `clear` for multiple inputs (at least 3) within one data-driven test. The test should check that the array is reset correctly.

(b) For each test case you made in (a), justify why you added each input value. Write this justification (and the comments for (d) in a pdf titled justifications).

(c) Repeat the tests from (a) with completely different Object types.

(d) When you changed Object types in tests from (a) to (c), did any of your motivation for the test input values you picked change? Highlight why or why not.

Submit: `MyArrayListTest.java`, `DataDrivenMyArrayList.java`, and justifications.pdf for grading.