

Curso > Week 4... > 7. Testi... > Exercis...

Audit Access Expires 5 de ago de 2020

You lose all access to this course, including your progress, on 5 de ago de 2020.

Exercise 1

Finger Exercises due Aug 5, 2020 20:30 -03 Completo

Exercise 1

1/1 point (graded)

ESTIMATED TIME TO COMPLETE: 4 minutes

Consider the following code specification:

```
def size(aSet):
"""
aSet is a collection of objects, which might be empty.
Objects are assumed to be of the same type.
"""
```

Here is a set of possible test cases to include in a black box test suite. Indicate which of the following conditions would make a good black box test suite for the function size by clicking on the appropriate choice(s).

Review: Black Box Test Suites

Black-box testing is a method of software testing that tests the *functionality* of an application. Recall from the lecture that a way to think about black-box testing is to look at both:

- The possible paths through the specification.
- The possible boundary cases.

Undoubtably many - if not all - of the listed tests look like they would be pretty good for testing the function <code>size</code>. However, we want you to think critically about the way <code>size</code> is specified - including possible boundary cases - and pick a set of tests that

adequately and fully tests all paths and boundary conditions. Be sure the set of tests you pick does not have extraneous, useless, or repetitive tests.

Empty set
✓ Set of size 1
Set of odd size
Set of even size
✓ Set of size greater than 1
Set whose size is a prime number

Explanation:

A good black box test suite would contain tests for the following conditions: Empty set, Set of size 1, and Set of size greater than 1.

Black-box testing is a method of software testing that tests the functionality of an application. Recall from the lecture that a way to think about black-box testing is to look at both the paths through the specification and the possible boundary cases. In this example, the boundary cases all have to do with the size of aSet. Specifically these boundary cases are when aSet contains zero, one, or many items.

The remaining conditions would not further test the functionality of the size function because an odd, even, or prime sized set are all sets of size greater than 1. Nothing in the function specification suggests there is anything special or unique about odd, even, or prime sized sets, so testing those cases specifically simply repeats the test "Set of size greater than 1".

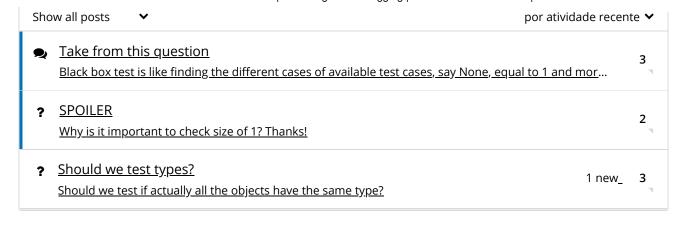
Enviar

1 Answers are displayed within the problem

Exercise 1

Ocultar discussão

Topic: Lecture / / Exercise 1



© All Rights Reserved