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 4

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

Exercise 4

1/1 point (graded)

ESTIMATED TIME TO COMPLETE: 4 minutes

Consider the following function definition:

```
def union(set1, set2):
    """
    set1 and set2 are collections of objects, each of which might be empty.
    Each set has no duplicates within itself, but there may be objects that
    are in both sets. Objects are assumed to be of the same type.

    This function returns one set containing all elements from
    both input sets, but with no duplicates.
    """
    if len(set1) == 0:
        return set2
    elif set1[0] in set2:
        return union(set1[1:], set2)
    else:
        return set1[0] + union(set1[1:], set2)
```

Assume that `union` is called with strings as arguments.

Please select the best glass box test suite for the function `union` from the following options:

- Test Suite A: `union('','')`, `union('','a')`, `union('','ab')`, `union('a','')`, `union('a','b')`, `union('c','ab')`, `union('de','')`, `union('ab','c')`, `union('cd','ab')`
- Test Suite B: `union('abc','')`, `union('abc','a')`, `union('abc','ab')`, `union('abc','d')`, `union('abc', 'abcd')`
- Test Suite C: `union('','abc')`, `union('a','abc')`, `union('ab','abc')`, `union('abc','abc')`
- Test Suite D: `union('','abc')`, `union('a','abc')`, `union('ab','abc')`, `union('d','abc')`

✔

Explanation:
A good glass box test suite would try to test a good sample of all the possible paths through the code. So, it should contain tests that test when `set1` is empty, when `set1[0]` is in `set2`, and when `set1[0]` is not in `set2`. The test suite should also test when the recursion depth is 0, 1, and greater than 1.
Remember that glass box testing is a method of software testing that tests the internal structures and workings of a piece of code. When we look at the code for `union`, we see a set of conditionals that ask about `set1`. Thus a good glass box test suite will contain tests that match the following lines from the conditional statements in the code:

- `len(set1) == 0` - matched by the test `union('', 'abc')`
- `set1[0] in set2` - matched by the test `union('a', 'abc')`
- `set1[0] not in set2` (this is the `else:` of the conditional) - matched by the test `union('d', 'abc')`

In addition, because `union` is a recursive function, we should make sure our test set considers a recursion depth of 0, 1, and many. In this case, recursion depth is covered by some of the tests we've already chosen:

- Recursion depth = 0 - matched by the test `union('', 'abc')`
- Recursion depth = 1 - matched by the tests `union('a', 'abc')`, `union('d', 'abc')`
- Recursion depth > 1 - matched by the test `union('ab', 'abc')`

Note that this test suite is NOT path complete because it would take essentially infinite time to test all possible recursive depths.

Let's examine now why the other test suites weren't as good as Test Suite D:

- Test Suite A looks at a good sampling of set sizes for `set1` and `set2`, but does not explore the if-else paths in the code. `set1` never contains any element in `set2`.
- Test Suite B does not explore the paths in the code because it never varies the contents of `set1`.
- Test Suite C does not contain a test that explores the path when `set1` has an element that is not in `set2`.

Enviar

Answers are displayed within the problem

Exercise 4

Ocultar discussão

Topic: Lecture 7 / Exercise 4

Show all posts por atividade recente

String indexing confusion
given someString = 'a' why does someString[1] produce an index error (because the value at index [1... 3 new

SPOILER
Why do we need to also check union('ab', 'abc')? Thanks! 3

SPOILER
Why don't we need to test if set 2 is empty? I chose the incorrect answer because it was the only choi... 3