



Class:

S7 ICT

Date:

Thursday, April the 20th, 2023

Teacher:

Mr Barsamian

B Test — With computer

Family name: _____

First name: _____

Duration: 20 minutes.

*This test has to be done both on paper and on computer.
At the end of the test, make sure to upload the python file
on the Teams assignment (or on your teacher's USB key).*

*Some questions are bonuses, and it is highly advised to
do them only at the end, when everything else has been
done.*

*If needed, the candidate can also handle some comments
inside the code or on paper.*

*Please keep track of the clock, and avoid spending too
much time on a question. Stay focused, and good luck!*



Short description of this work:

This test is the continuation of the BTest from yesterday.

We will handle strings using a Tree data structure. The start of a string is the root of the Tree, and then we will figure out different ways to encode them.

1 Introduction

0 points

Please start by downloading the following file (or continue with the file from yesterday), that you'll have to update for this test:

http://www.barsamian.am/2022-2023/S7ICTB/BTest_Words.py

A string consists of multiple characters. We will model the strings seen in this test by trees, see Listing 1. If a tree models a string, then each of its nodes will either be empty, either will contain a single character `data` and have two children `left` and `right`.

```

1 class Tree:
2     def __init__(self, data, left=None, right=None):
3         self.data = data
4         self.left = left
5         self.right = right
6
7     def __str__(self):
8         return str(self.data)

```

Listing 1: Python code for the Tree data structure.

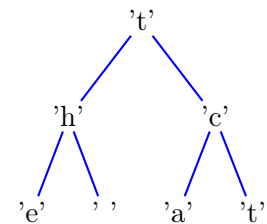
The file contains Listing 2, which implements the tree on the right.

```

1 str1_node4 = Tree('e')
2 str1_node5 = Tree(' ')
3 str1_node6 = Tree('a')
4 str1_node7 = Tree('t')
5 str1_node2 = Tree('h', str1_node4, str1_node5)
6 str1_node3 = Tree('c', str1_node6, str1_node7)
7 str1_node1 = Tree('t', str1_node2, str1_node3)

```

Listing 2: Python code for the first string.



2 Still counting

3 points

1. We now define the function `string_representation` which is the function `mystery3`, see Listing 3. This means that the string associated to a tree is the result of the function `mystery3` on that tree. Yesterday, we created a function `count_spaces` that takes as argument a tree and that counts the number of spaces in that tree. Whether or not you successfully implemented this function is irrelevant for this question: you can assume that you have a function that does this task (for instance, create a mock function that always return 1 and just pretend it is the correct result).

Please use the function `count_spaces` to create a new function `count_words` that takes as argument a tree and that counts the number of words in the string associated to this tree.

Unit tests:

- `count_words(str1_node1)` should return 2;
- `count_words(str2_node1)` should return 3.

```

1 def mystery3(tree):
2     if tree == None:
3         return ""
4     return tree.data + mystery3(tree.left) + mystery3(tree.right)

```

Listing 3: Python code for the mystery function.

2. A tree is correctly representing a string if all its nodes have a single character in the `data` field. All trees given previously are correct, but for instance the tree defined by `Tree('the cat')` is not correct.

Please write a function `is_correct` that takes as argument a tree and that returns `True` if this tree is correct, and that returns `False` otherwise.

BONUS — The data field is not necessarily a string. Verify that your function works is, for instance, the data field contains a number.

BONUS Please write a function `is_really_correct` that takes as argument a tree and checks that the root node contains a capital letter (uppercase), and that all other nodes contain either a space, either a letter (lowercase or uppercase), either punctuation marks (! ? , ; : - () " ' .).