



Class:

S7 ICT

Date:

Wednesday, April the 19th, 2023

Teacher:

Mr Barsamian

B Test — With computer

Family name: _____

First name: _____

Grade: ___ / 10
(only 7 marks today, the rest tomorrow)

Duration: 45 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 your teacher's USB key.*

*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:

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, that you'll have to update for this test:

http://www.barsamian.am/2022-2023/S7ICTE/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.

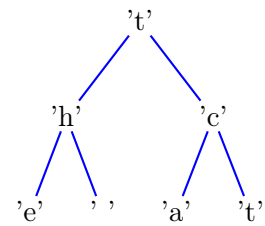
The file contains Listing 2, which implements the tree on the right.

```

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 First steps

5 points

1. The file also contains Listing 3. Please draw, on paper, a tree which would correspond to this implementation.

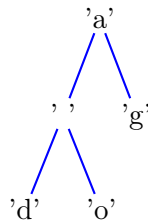
```

1 str2_node8 = Tree('s')
2 str2_node9 = Tree(' ')
3 str2_node10 = Tree('s')
4 str2_node11 = Tree(' ')
5 str2_node4 = Tree('i', str2_node8, str2_node9)
6 str2_node5 = Tree('i', str2_node10, str2_node11)
7 str2_node6 = Tree('t')
8 str2_node7 = Tree('!')
9 str2_node2 = Tree('h', str2_node4, str2_node5)
10 str2_node3 = Tree('i', str2_node6, str2_node7)
11 str2_node1 = Tree('t', str2_node2, str2_node3)

```

Listing 3: Python code for the second string.

2. Let us consider the tree drawn below. Please write, in the Python file, a Tree which would model this string.



3. In the Python file, the functions `mystery1`, `mystery2` and `mystery3` are provided (also given in Listing 4) which take as argument a tree. Please:
 - (a) explain step by step what happens if you execute `mystery1(str1_node3)`;
 - (b) explain step by step what happens if you execute `mystery2(str1_node3)`;
 - (c) explain step by step what happens if you execute `mystery3(str1_node3)`.

```

1 def mystery1(tree):
2     if tree == None:
3         return ""
4     return mystery1(tree.left) + mystery1(tree.right) + tree.data
5
6 def mystery2(tree):
7     if tree == None:
8         return ""
9     return mystery2(tree.left) + tree.data + mystery2(tree.right)
10
11 def mystery3(tree):
12     if tree == None:
13         return ""
14     return tree.data + mystery3(tree.left) + mystery3(tree.right)

```

Listing 4: Python code for the mystery functions.

3 Counting

2 points

1. Please give a function `count_nodes` that takes as argument a tree and that counts the number of nodes in that tree. We only count the nodes that are drawn on the representations on the previous pages, we don't count `None`.

Unit tests:

- `count_nodes(str1_node1)` should return 7;
- `count_nodes(str2_node1)` should return 11.

2. Please give a function `count_spaces` that takes as argument a tree and that counts the number of spaces in that tree.

Unit tests:

- `count_spaces(str1_node1)` should return 1;
- `count_spaces(str2_node1)` should return 2.