MIDDLE EAST TECHNICAL UNIVERSITY
DEPARTMENT OF COMPUTER ENGINEERING

CENG 111

Fall 2024

Take-Home Exam 4

REGULATIONS

Due date: 23:59, 18 January 2025, Saturday *(Not subject to postpone)*

Submission: Electronically. You should save your program source code as a text file named `the4.py`. Check the announcement on the ODTUCLASS course page for the submission procedure.

Team: There is **no** teaming up. This is an EXAM.

Cheating: Source(s) and Receiver(s) will receive zero and be subject to disciplinary action.

INTRODUCTION

In this Take-Home Exam (THE), the task is to discover the sequences of operations performed on some strings. To keep things simple, we will work with strings consisting of letters `'o'` and `'x'`, and hence, call such a string an *OX string*.

More formally, an OX string is defined as a string (S) of letters where the i^{th} element of the string (S_i) is either `'o'` or `'x'`. On an OX string S, we define an operation, $\Psi(S)$, which produces a modified string, as follows:

- All `'x'` letters remain the same.
- A randomly chosen `'o'` letter is converted to an `'x'`.
- All the other `'o'` letters remain the same.

As an example, the operation Ψ(`'oooxoo'`) may produce the following OX strings as outcomes: `'xooxoo'`, `'oxoxoo'`, `'ooxxoo'`, `'oooxxo'`,`'oooxox'`.

PROBLEM & SPECIFICATIONS

You are expected to write a function `OX_to_tree(L)` where `L` is an unordered list of OX strings of the same length. `OX_to_tree(L)` is expected to form and return an n-ary tree where each child is a valid Ψ operation of its parent.

- The tree to be returned should follow the representation below:
  if $terminal(tree)$ **then** $datum(tree)$
  else `[`$datum(tree)$`,` $child_1(tree)$`,` $child_2(tree)$`,...]`
- It is quite possible that there are multiple solutions. `OX_to_tree(L)` will return any one of the possibilities.

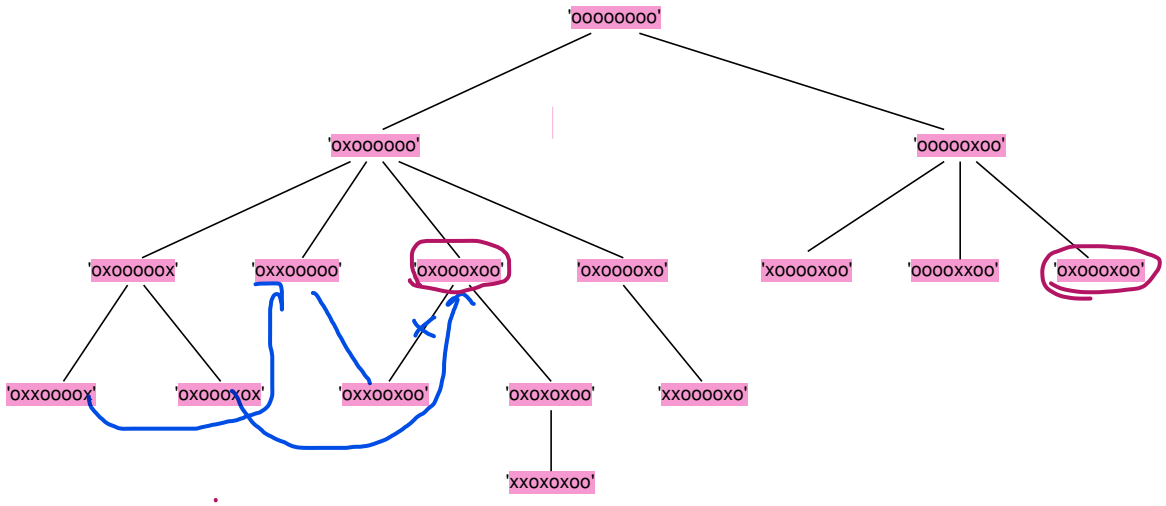

Figure 1: An example tree illustrating a set of Ψ operation applications for a starting OX string.

- It is important that all elements of the list are placed on the tree (i.e., no string will be left out of the tree). Moreover, it is guaranteed that there is a solution and the provided OX string list forms a tree (as described above).
- It is possible that an element is repeated in the list. Though you are not allowed to make them sibling nodes.
- Ψ is defined only on an OX string that has at least one `'o'` present.
- The root can be any OX string.

SAMPLE RUN

For the example drawn in Figure 1, your function should work as follows:

```
>>> Misal = ['xooooxoo','oxxooooo','oxooooxo','oooooxoo','xxoxoxoo',
             'oxoxoxoo','oxoooxoo','oxooooox','oxoooxoo','ooooxxoo',
             'xxoooooxo','oxxooooox','oooooooo','oxoooxox','oxxooxoo',
             'oxooooxo']

>>> print(OX_to_tree(Misal))
['oooooooo',
 ['oxooooooo',
  ['oxooooox', 'oxxooooox', 'oxoooxox'],
  'oxxooooo',
  ['oxoooxoo', 'oxxooxoo', ['oxoxoxoo', 'xxoxoxoo']],
  ['oxooooxo', 'xxooooxo']]
 ['oooooxoo', 'xooooxoo', 'ooooxxoo', 'oxoooxoo']]
```

(The output is formatted by us with indentations to enhance understandability.)

Your implementation might yield a different result, which is entirely possible due to variations

in child ordering and/or parent-child assignments. The sole criterion for validity is adherence to the rules outlined in the "PROBLEM & SPECIFICATION" section above.

RESTRICTIONS & GRADING

- You are free to use recursion or iteration, and define as many functions as necessary.
- You are not allowed to import any libraries.
- A set of arbitrarily selected students will be subject to oral examination about their solutions. The details will be announced on ODTUclass later.
- Your program will be graded through an automated process and therefore, any violation of the specifications will lead to errors (and reduction of points) in automated evaluation. You should especially avoid printing something on screen.
- Your solutions will not be tested with incorrect or erroneous inputs.
- Your program will be tested with multiple data (a distinct run for each data). Any program that performs only 30% and below will enter a glass-box test (eye inspection by the grader TA). The TA will judge an overall grade in the range of [0,30]. The glass-box test grade is neither open to discussion nor explanation.
- A program based on randomness will be graded zero.