

INSTRUCTIONS

- You have 10 minutes to complete this quiz.
- The exam is closed book, closed notes, closed computer, closed calculator.
- The final score for this quiz will be assigned based on **effort** rather than correctness.
- Mark your answers **on the exam itself**. We will *not* grade answers written on scratch paper.
- For multiple choice questions,
 - means mark **all options** that apply
 - means mark a **single choice**

Last name	
First name	
Student ID number	
CalCentral email (_@berkeley.edu)	
Teaching Assistant	<input type="radio"/> Alex Stennet <input type="radio"/> Kelly Chen <input type="radio"/> Angela Kwon <input type="radio"/> Michael Gibbes <input type="radio"/> Ashley Chien <input type="radio"/> Michelle Hwang <input type="radio"/> Joyce Luong <input type="radio"/> Mitas Ray <input type="radio"/> Karthik Bharathala <input type="radio"/> Rocky Duan <input type="radio"/> Kavi Gupta <input type="radio"/> Samantha Wong
Name of the person to your left	
Name of the person to your right	
<i>All the work on this exam is my own.</i> (please sign)	

1. (5 points) Tree Time

For each line in the implementation of the `IterableTree` class below, fill in the square to the left of the line if removing will help pass the doctests **and the implementation contains as few lines of code as possible**. Don't cross out any docstrings or doctests.

The `__iter__` generator for this class should yield the values of the tree starting with the root, and yield all of the values of the left branch before any values of the right branch. The `Tree` class definition is shown to the right.

- | | |
|---|--|
| <input type="checkbox"/> <code>class IterableTree:</code>
<input type="checkbox"/> <code>class IterableTree(Tree):</code>
<input type="checkbox"/> <code> def __init__(self, root, branches=[]):</code>
<input type="checkbox"/> <code> Tree.__init__(root, branches)</code>
<input type="checkbox"/> <code> Tree.__init__(self, root, branches)</code>
<input type="checkbox"/> <code> def __iter__(self):</code>
<input type="checkbox"/> <code> """Yield the entries of this tree.</code>

<input type="checkbox"/> <code> >>> T = IterableTree</code>
<input type="checkbox"/> <code> >>> t = T('A', T(2, T('C'), T(4)), T('E', None, T(6)))</code>
<input type="checkbox"/> <code> >>> list(t)</code>
<input type="checkbox"/> <code> ['A', 2, 'C', 4, 'E', 6]</code>
<input type="checkbox"/> <code> """</code>

<input type="checkbox"/> <code> yield self.root</code>
<input type="checkbox"/> <code> yield root</code>
<input type="checkbox"/> <code> for branch in self.branches:</code>
<input type="checkbox"/> <code> if branch:</code>
<input type="checkbox"/> <code> if self.branch:</code>
<input type="checkbox"/> <code> branch = iter(branch)</code>
<input type="checkbox"/> <code> for root in branch:</code>
<input type="checkbox"/> <code> for root in branch():</code>
<input type="checkbox"/> <code> yield self.root</code>
<input type="checkbox"/> <code> yield root</code>
<input type="checkbox"/> <code> yield self.root</code>
<input type="checkbox"/> <code> yield root</code> | <pre>class Tree: def __init__(self, root, branches=[]): self.root = root self.branches = branches def is_leaf(self): return not self.branches</pre> |
|---|--|