

CSC 148: Introduction to Computer Science

Week 8

Tree deletion

Mutating Trees



University of Toronto Mississauga,
Department of Mathematical and Computational Sciences



Delete an Item

```
def delete_item(self, item) -> bool:
    """Delete *one* occurrence of <item> from this tree.

    Return True if <item> was deleted, and False otherwise.
    """
```



Implementation Starter Code

```
def delete_item(self, item) -> bool:

    if self.is_empty():

        return False    # item isn't in the tree

    elif self._root == item:

        self._delete_root()

        return True     # item was deleted

    else:

        ... # recurse somehow
```



Empty Trees

```
>>> t = Tree(10, [Tree(1, []), Tree(2, []), Tree(3, [])])
>>> t.delete_item(1)
True
>>> t.delete_item(2)
True
>>> t.delete_item(3)
True
```



A Hidden Assumption



- “self._subtrees doesn't contain any empty trees.”



A Representation Invariant!

- State assumptions using a RI:
 - “self._subtrees doesn’t contain any empty trees.”



Reminders and Tips

- Revisit readings, especially re-read the notes on mutating trees!
- Run the code and add tests to convince yourselves
- Partially trace through the recursive step
- Analyze: what happens if we have `_subtrees` contain empty trees?
what implications are there? etc..
- Ask yourselves questions, try out things, test, trace, analyze, etc.
- If you have anything confusing or unclear, formulate clear questions and come talk to us!