Assignment 2: Design Patterns

(may be done by a team of at most two students)
Assigned: Wednesday, September 30, 2019
Due: Friday, October 16, 2019 (11:59 pm)
Note: The mid-term exam is on Thurs, October 22.

Part 1: Generic Tree Iterators
(part 2 to be assigned)

The file `GenericIterators.java` posted under `Resources → Assignments` defines generic versions of `AbsTree`, `Tree`, and `DupTree` discussed in class. The file also gives the outlines of generic external iterators for these classes, called `AbsTreeIterator`, `TreeIterator` and `DupTreeIterator` respectively.

Considerable code-factoring can be achieved in their definitions because `TreeIterator` and `DupTreeIterator` only need to define their constructors; the entire logic of traversal can be kept in `AbsTreeIterator`.

Also given in `GenericIterators.java` are tester methods that represent sets and bags as trees and duptrees, respectively, and carry out *containment* tests by invoking the `boolean` method `contains`, to be defined by you in this assignment.

What you should program:

(i) Complete the `AbsTreeIterator` class by writing code for its constructor and the methods `next()`, `hasNext()` and `stack_tree_nodes()`. Note that, for duptrees, the `next()` method should return the value in a `DupTree` node as many times as specified by the count associated with this node. Each invocation of `next()` returns only one value.

Note: Refer to Lecture 6 slide #35 for guidance on how to define the private method `stack_tree_nodes()`.

(ii) Complete the definition of the static `boolean` method `contains(AbsTree<T> tr1, AbsTree<T> tr2)` in class `GenericIterators` so that it works for sets as well as bags. For your reference:

- A set s1 contains set s2 if every member of s2 is also a member of s1.
- A bag b1 contains bag b2 if every member, x, of b2 is also a member of b1; also, the number of occurrences of x in b2 is less than or equal to the number of occurrences of x in b1.

Important: There are two key requirements for `contains`:

(i) that the test is carried out by making only one traversal through each set/bag; and
(ii) that the test returns `false` as soon as possible, i.e., without necessarily traversing the entire set/bag.

These requirements can be met because the elements of sets and bags are `Comparable` and therefore can be enumerated in order.

Run the five test methods given in `GenericIterators.java`. Each test should print out on the console the values that are compared during their execution in to clarify their behavior. Check your output against the answers given in `A2_Part1_output.txt`.

What to Submit. Prepare a top-level directory named *A2_Part1_UBITId1_UBITId2* if the assignment is done by a team of two students; otherwise, name it as *A2_Part1_UBITId* if the assignment is done solo. (Order the *UBITId*s in alphabetic order in the former case.) In this directory, place your source file `GenericIterators.java`. No diagrams are required. Compress the directory and submit the compressed file using the `submit_cse522` command.

End of Assignment 2 Part 1