Assignment 2

(may be done by a team of at most two students)
Assigned: Monday, September 23, 2019
Due: Wednesday, October 9, 2019

Part 1: Generic External Iterator Design Pattern

(Part 2 will be assigned later)

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 definition 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 four tester methods that represent sets and bags as trees and duptrees respectively and carry out *disjointness* and *containment* tests by invoking two boolean methods, `disjoint` and `contains`, to be defined by you in this assignment.

Your tasks are as follows:

(i) Complete the `AbsTreeIterator` class by writing code for its constructor as well as the methods `next()`, `hasNext()` and `stack_left_spine()`. 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. Reference: Lecture 7 slides.

(ii) Complete the definition of the static `boolean` method `disjoint(AbsTree<T> tr1, AbsTree<T> tr2)` in class `GenericIterators` so that it works for sets as well as bags. The tester methods `test1` and `test2` create, respectively, two sets and two bags and invoke `disjoint`. For your reference:

- A set s1 is disjoint from set s2 if they have no members in common. Similarly,
- A bag b1 is disjoint from bag b2 if they have no members in common.

(iii) 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. The tester methods `test3` and `test4` create, respectively, two sets and two bags and invoke `contains`. 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: For both `disjoint` and `contains`, there are two key requirements:

(i) that these tests are carried out by making only one traversal through each set/bag; and
(ii) that these tests return `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. You should print out on the Console the values that are compared during their execution in to clarify their behavior. The desired output is shown in file `A2_Part1_Console_Output.txt`.

Run `GenericIterators.java` to completion and check that your console output agrees with the desired output.

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