

Homework # 8

due Monday, March 30th, 11:00 AM

1 Integer Sets

In this homework, you will implement three ways to represent a set of integers. We provide a solution to Exercise 13 of Chapter 13 (page 238). In this Homework, you will modify this code so that we have an interface `IntSet`, and abstract base class (put keyword `abstract` before the class declaration) `AbstractIntSet` that implements this interface and provides default (possibly inefficient) implementations of most of the methods, and three concrete classes that inherit from the abstract base class and provide actual implementations:

TreeIntSet This class will use the BST implementation from the Exercise 13 solution. You will need to implement the iterator class: the local class will need to have its own linked list of BST nodes to keep track of nodes that have not yet been visited (essentially a stack, but do not use the library class `Stack`).

LinkedIntSet This class will use linked lists to implement the functionality.

HashedIntSet This class will use a `java.util.HashMap` which maps `Integer` objects to themselves (if in the set) or `null` (if not in the set). The `keys` method of this class which returns a `java.util.Enumeration`, which will be very useful. (Don't use the library class `HashSet`.)

The `IntSet` interface requires the following functionality:

find Return true if the parameter is in the set.

add Add the parameter to the set if it is not already there.

equals Return true if the parameter implements `IntSet` and has the same elements.

hashCode Returns the sum of the elements.

includedIn Returns true if all this set's elements are in the parameter set.

iterator Return an object of a class that implement `IntIterator`.

toString Return a string for the set.

The `IntIterator` interface requires

hasNext True if `next()` will return an integer.

next Return the next integer in the set, or throw an exception.

Both these interfaces (and the solution to Exercise 13) are available for you in

`/afs/cs.uwm.edu/users/classes/cs431/src/homework8`

The interfaces should *not* be changed.

You need to give (perhaps inefficient) implementations of all methods except `add` and `iterator` in the `AbstractIntSet` class. The inefficient implementation should be overridden if a more efficient implementation is possible.

We also provide a driver and some skeleton files.

2 Submitting Your Work

Leave your code in the directory `homework8` in your AFS folder:

- `AbstractIntSet.java`
- `TreeIntSet.java`
- `LinkedIntSet.java`
- `HashedIntSet.java`