CIS 181 - Lab 4

Class Development

Overview

You are asked to work with sets of non-negative integers from 0...100 with effective space/time methods. A set of integers is stored as an array of boolean. The array is indexed from 0…100. The value provided at the position in the array indicates whether a particular integer is a member of the set. Note that a set does not allow duplicates.

Example

To demonstrate, look at a smaller example, say sets of non-negative integers from 0…4. We create the empty set, as shown in Figure 1, indicating that none of the values are in the set, thus every cell contains *false*.

To add an integer from 0...4 to a set, we need to change the value held in that cell to *true*. The cell index is the value we wish to have in the set. For instance, Figure 2 show the set {2, 4}. Notice cell 2 is set to *true* and cell 4 is set to *true* while the other cells are still *false*.

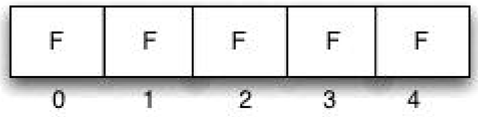

Figure 1

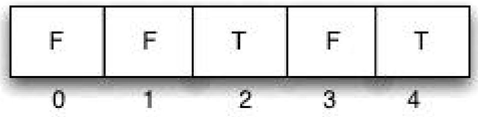

Figure 2

You are asked to develop a class called *IntSet* where any instance of *IntSet* represents a set of integers from 0…100. Since we are developing a class for set representation, it should allow the usual set operations. Provided is a file with the structure for the class, including the assertions for each method.

Goals

- Write methods according to preconditions and postcondtions.
- Prepare for requirements review.
- Resolve problem understanding issues prior to task initiation by formulating appropriate questions.

Tasks

1. You may form a group with your classmates, and review tasks with questions and answers.
2. Individually meet with the instructor/TA to resolve issues.
3. In Eclipse, create a new Java project called "Lab 4", and a new package called "set" under the "src" directory. Import the 2 files (IntSet.java and TestIntSet.java) into package "set".
4. Complete the following methods in the IntSet class
    1. public boolean remove(int element)
    2. public IntSet union(IntSet s)
    3. public IntSet intersection(IntSet s)
    4. public IntSet complement(IntSet s)

5. During the method implementation, you need to design **test cases** to verify if your implementation is correct. Test cases should be recorded in the file TestIntSet.java. When completed, both the source files *IntSet.java* and *TestIntSet.java* must be uploaded to myCourses.