CSSE221: Fundamentals of Software Development Honors

Homework 2 Solutions

Programming assignment

Finish BigRational, as stated in the specification, being sure you have checked in your final, fully documented copy to the repository.

Written homework to do before week 2 (due before beginning of class next Monday).

As usual, please bring hardcopy for written answers to class to hand in on Monday.

2. I want to compare two BigRationals, x and y, to see if they are equal value (intuitively, x == y).

    a. Write an expression for this using the compareTo() method.

        if (x.compareTo(y) = = 0)

    b. Write an expression for this using the equals() method.

        if (x.equals(y))

    c. What is compared if instead I write (x == y)? (Hint, you saw this in section 2.2 if you read the book. This is important to understand!)

        This compares memory addresses, not values in the objects. Checking whether or not it's the same object, not if it has the same value.

3. For the sake of simplicity, on this question, you may ignore effects such as operating system interrupts that may change the runtime slightly. (Focus on the meaning of big-oh only.)

    a. A block of code is characterized as O(n) and runs in 23 milliseconds. If n is increased by a factor of 8, what is the new runtime of the code?

        184 milliseconds

    b. A block of code is characterized as O(n2) and runs in 400 milliseconds. If n is increased by a factor of 3, what is the new runtime of the code?

        3,600 milliseconds

4. What combination of control structure(s) gives code that is O(n3)?

    Triple nested loops

5. How many times is the sum modified (in terms of *n*):

```
for (int i = 0; i < n; i++) {
  for (int j = 0; j < n; j++) {
        sum++;
  }
  for (int j = 0; j < n; j++) {
        sum++;
  }
}
```

Exact answer = ___$2n^2$___, giving an asymptotic answer of O(___n^2___). (5 Points Each)

6. Write the code for a method that does the following. (Note, you may choose to type this into Eclipse to test it, but you aren't required to do so. If you do type it in, please still print it out.)

Pass into the method a BigRational object. Evaluate the object to determine if it is equal to its own absolute value. If it is, print the message "Good going", and terminate the program (or at least exit the method). If it is not, print the message "Not good going", and terminate the program. Regardless of the outcome of this test, print the message "Done" before terminating.

Write this code twice: once using if-then-else block(s) to evaluate the data and handle a possible error, and once using a try-catch block to evaluate the data and handle a possible error. When using a try-catch block, use a "throw" statement to throw an exception for each case and handle that case within the catch block; you are allowed to use an if statement within the try-catch, but you should not print within the if statement. (20 Points Each)

Note: This code would never be written using a try-catch block in a production environment, nor do we typically both throw and catch exceptions in the same method; we do it here to make sure you understand important structures.

```
public void equalToAbsoluteValueIf(BigRational br) {
        if (br!=null) {
                if (br.equals(br.abs())) {
                        System.out.println("Good going");
                } else {
                        System.out.println("Not good going");
                }
        }
        System.out.println("Done");
}
```

(continue on next page)

```
public void equalToAbsoluteValueTry(BigRational br) {
        try {
                if (br.equals(br.abs())) {
                        throw new Exception("Good going");
                } else {
                        throw new Exception("Not good going");
                }
        }
        catch (NullPointerException e) {}
        catch (Exception e) {
                System.out.println(e.getMessage());
        } finally {
                System.out.println("Done");
        }
}
```