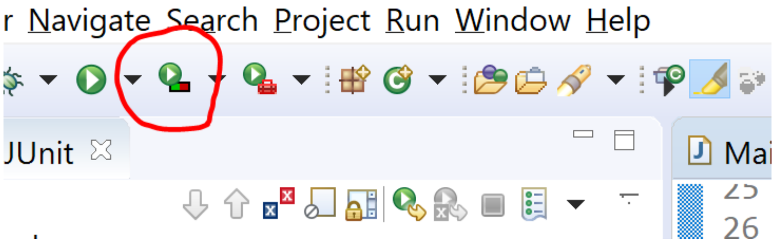
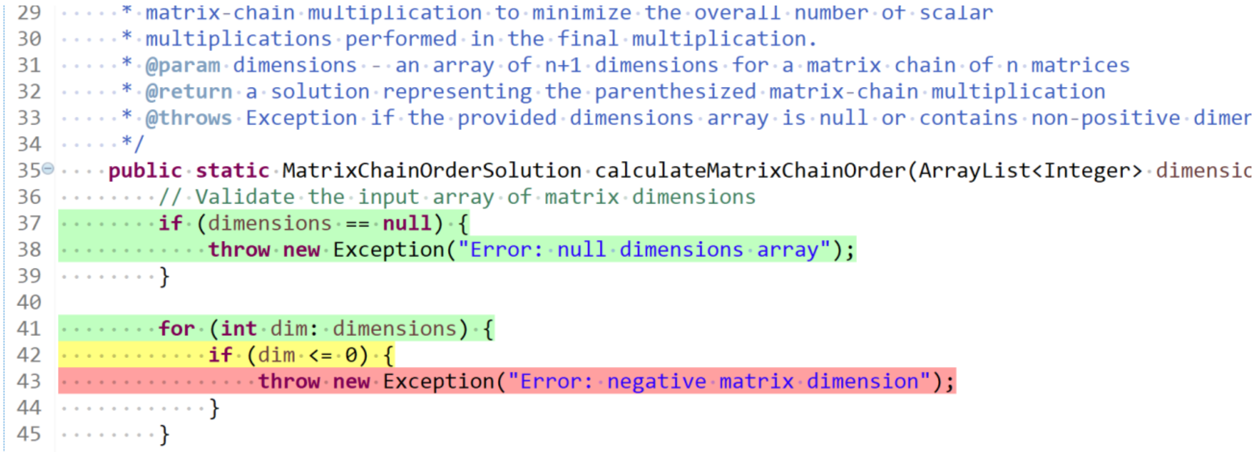
Getting Started with Assignment 3b

COMP 550.001 Fall 2017

This walks you through code coverage in Eclipse. To run the tests with code coverage, use the highlighted button below:



When you first run the provided unit tests, you will see that lines of code in the MatrixChainOrder.java file are highlighted. (You can ignore code coverage coloring in the unit tests file).

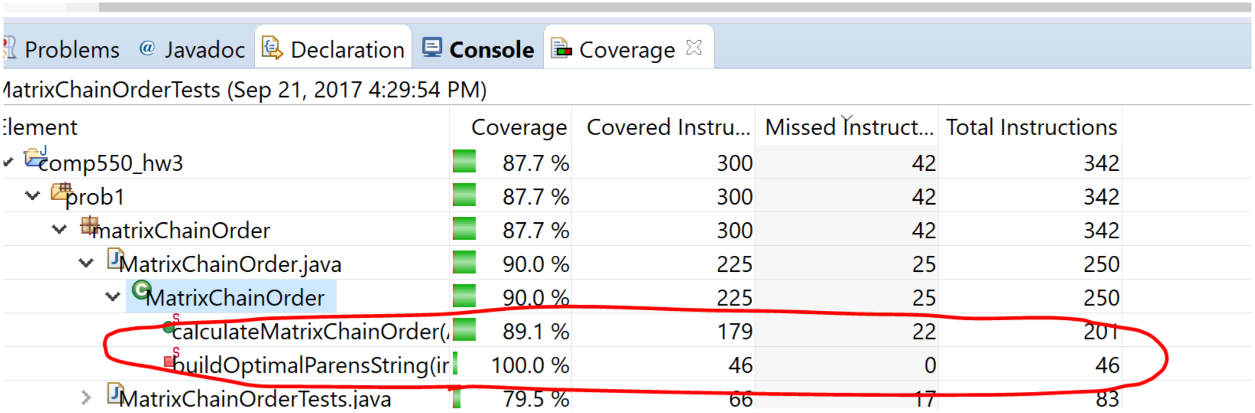


```
     * matrix-chain multiplication to minimize the overall number of scalar
     * multiplications performed in the final multiplication.
     * @param dimensions - an array of n+1 dimensions for a matrix chain of n matrices
     * @return a solution representing the parenthesized matrix-chain multiplication
     * @throws Exception if the provided dimensions array is null or contains non-positive dimer
     */
    public static MatrixChainOrderSolution calculateMatrixChainOrder(ArrayList<Integer> dimensic
        // Validate the input array of matrix dimensions
        if (dimensions == null) {
            throw new Exception("Error: null dimensions array");
        }

        for (int dim: dimensions) {
            if (dim <= 0) {
                throw new Exception("Error: negative matrix dimension");
            }
        }
```

Green lines are those that are completely covered. For an if statement, for example, this means the if was evaluated both to true and false during the tests' executions. A yellow line was only partially evaluated. For an if statement, this means it either always evaluated to true, or always evaluated to false. A red line was never hit.

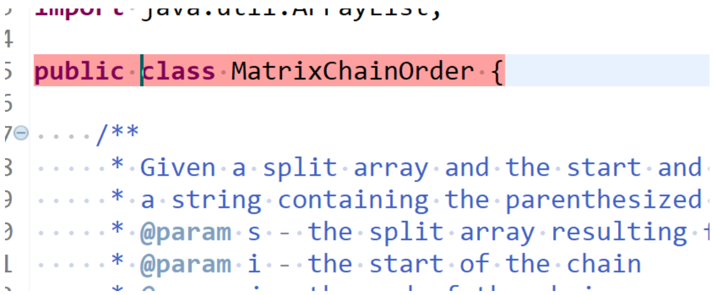
To get full credit for this problem, you must have all lines of `buildOptimalParensString` and `calculateMatrixChainOrder` at 100% unit test code coverage. Note that you can also check code coverage at the bottom in the Coverage window.



latrixChainOrderTests (Sep 21, 2017 4:29:54 PM)

lement	Coverage	Covered Instru...	Missed Instruct...	Total Instructions
comp550_hw3	87.7 %	300	42	342
prob1	87.7 %	300	42	342
matrixChainOrder	87.7 %	300	42	342
MatrixChainOrder.java	90.0 %	225	25	250
MatrixChainOrder	90.0 %	225	25	250
calculateMatrixChainOrder(	89.1 %	179	22	201
buildOptimalParensString(ir	100.0 %	46	0	46
MatrixChainOrderTests.java	79.5 %	66	17	83

Finally, note that due to the way Eclipse is checking code coverage, you can have the functions completely covered, but the class itself still marked as not covered. This is expected, and should be the only red-highlighted line in the file for full credit:



```
import java.util.ArrayList;

public class MatrixChainOrder {

    /**
     * Given a split array and the start and
     * a string containing the parenthesized
     * @param s - the split array resulting
     * @param i - the start of the chain
```

If at any time you want to clear out the coverage coloring, this button above the console will do it: