CS2040 Data Structures and Algorithms: Problem Set 0

Due: Fri, Aug 24, 2018

Harold Soh

Ungraded: Question scores will not count towards your final class grade. However, completion of this problem set gets you participation points.

Collaboration Policy. You are encouraged to work with other students on solving these problems. However, you **must** write up your solution **by yourself**. We may, randomly, ask you questions about your solution, and if you cannot answer them, we will assume you have cheated. In addition, when you write up your solution, you **must** list the names of every collaborator, that is, every other person that you talked to about the problem (even if you only discussed it briefly). Any deviation from this policy will be considered cheating, and will be punished severely, including referral to the NUS Board of Discipline.

Quick Start

In CS2040, we will be writing programs in Java using IntelliJ, an integrated development environment (IDE) consisting of editing, debugging, and performance measurement tools. Problem Set Zero focuses on getting us ready for the semester. Your tasks this week:

- Set up IntelliJ for writing Java programs.
- Write a sample program using IntelliJ.
- Submit your code to Coursemology.

Task 1: Set up IntelliJ and Coursemology

Let's get IntelliJ installed and running on your system.

- If you have not yet installed **Java 8 JDK**, download and install it from `http://www.oracle.com/technetwork/java/javase/downloads/index.html`
- Download and install **IntelliJ Community** from `https://www.jetbrains.com/idea/download/`. You can learn more about IntelliJ via the documentation and video tutorials at `https://www.jetbrains.com/idea/documentation/`.

If you're still not on `https://coursemology.org`, check your email inbox (and spam folder) for the invitation email. On Coursemology, click on Assessments on the left menu panel and click Problem Set 0 to start the exercise.

Task 2: Mystery Code

Let's create a simple project and get started with some code! Launch IntelliJ, and on the welcome window, click Create a New Project. Choose Java from the left panel and click Next (de-select any additional libraries and frameworks). Also make sure that the Project SDK shows a valid Java version. If not (e.g., you are using your laptop), you will have to configure your project SDK by pressing New and navigating to where you installed your JDK.
Un-check Create a project from template and click Next. Enter the following details:

- Project Name: PS0
- Project Location: [Pick something sensible here or leave it as the default]

Click Finish and IntelliJ will build a starter project for you. Expand the PS0 directory (in the left panel) and you should see a directory called src. Right click on src and Create a new file called `Main.java` by going to New → Java Class. In the Dialog window that opens up, enter:

- Name: Main.java
- Kind: Class

and click OK. Let's modify `Main.java` with the following code which includes a new function called `MysteryFunction` and code to call this function from `main`.

Task 2: Mystery Code continued on next page. . .

```java
public class Main {

    static int MysteryFunction(int argA, int argB)
    {
        // what do you think Mystery Function does?
        // give an answer in Coursemology (Question 2)
        int c = 1;
        int d = argA;
        int e = argB;
        while (e > 0)
        {
            if (2*(e/2) !=e)
            {
                c = c*d;
            }
            d = d*d;
            e = e/2;
        }
        return c;
    }

    public static void main(String[] args) {
        int output = MysteryFunction(5, 5);

        System.out.printf("The answer is: " + output + ".");
    }
}
```

Figure 1: Main.java with our Mystery Function.

Run your program by right clicking on Main in the src folder and click Run Main.main(). In the future you can press the the play button ▶ on the top menu.

Problem: What is Mystery function doing? Try changing the arguments and see if you can figure it out. Give a short description, along with pre and post conditions and computational complexity (Question 2 in Coursemology).

Task 3: Submission!

In this final Task, we'll submit the code for your program onto Coursemology. Paste your code into the box in Coursemology (Question 3) and click Run Code to see if it passes the tests. You don't have to modify the code in this problem sheet; the sample code above will work just fine. In general, there are *private* test cases in addition to the visible public tests. Passing the public tests does not guarantee that your code will pass the private tests. Once you are happy with your code, finalize your submission.

— END OF PROBLEM SET ZERO —