

CS 570: Homework Assignment 2

Due: Feb 26th, 11:59pm

1 Assignment Policies

Collaboration Policy. Homework will be done individually: each student must hand in their own answers. It is acceptable for students to collaborate in understanding the material but not in solving the problems or programming. Use of the Internet is allowed, but should not include searching for existing solutions.

Under absolutely no circumstances code can be exchanged between students. Excerpts of code presented in class can be used.

Assignments from previous offerings of the course must not be re-used. Violations will be penalized appropriately.

Your code must include a comment with your name.

2 Assignment

This assignment asks you to implement a number of methods for a class `Complexity`. These methods should be implemented **using for loops**, as seen in class (except for the extra-credit one). In addition, each of these methods should print out the value of an accumulator that counts the number of “operations” performed. The notion of “operation” should be taken loosely; the idea is that if you are requested to implement a method of time complexity $\mathcal{O}(n)$, then it should print out values from 1 to n (or close enough). For example, the following code implements a method that has time complexity $\mathcal{O}(n)$:

```
void public method0(int n) {  
2   int counter=0;  
   for (i=0; i<n; i++) {  
4       System.out.println("Operation "+counter);  
       counter++;  
6   }  
}
```

The methods you should implement are:

- `public static void method1(int n)`: a method that has time complexity $\mathcal{O}(n^2)$.

- `public static void method2(int n)`: a method that has time complexity $\mathcal{O}(n^3)$.
- `public static void method3(int n)`: a method that has time complexity $\mathcal{O}(\log n)$.
- `public static void method4(int n)`: a method that has time complexity $\mathcal{O}(n \log n)$.
- `public static void method5(int n)`: a method that has time complexity $\mathcal{O}(\log \log n)$.
- (Optional) `public static int method6(int n)`: a method that has time complexity $\mathcal{O}(2^n)$. For this method you should consider using recursion. This can give you extra points if you miss any points from the previous methods, but the cap for the assignment is 100pts. You can not exceed 100pts with answering this extra-credit question.

3 Submission instructions

Your source code file has to be named Complexity.java. Compress this file into a zip file, and rename it to hw2_<surname>.zip. Submit the zip file mentioned above through Canvas.

No report is required.