EECS2030 Fall 2019
Lab 8
Analyzing the Time Complexity of a Program

Chen-Wei Wang

Due Date: 11:59 PM, Wednesday, December 4, 2019

- Texts in blue are hyperlinks to the corresponding documents/recordings.
- Students in the same lab section are allowed to work in groups of 2 or 3 students.

1 Policies

- **Your (submitted or un-submitted) solution to this lab exercise (which is not revealed to the public) remains the property of the EECS department. Do not distribute or share your code in any public media (e.g., a non-private Github repository) in any way, shape, or form. The department reserves the right to take necessary actions upon found violations of this policy.**
- When you submit your lab, you claim that it is **solely** your work. Therefore, it is considered as **an violation of academic integrity** if you copy or share **any** parts of your Java code during **any** stages of your development.
- When assessing your submission, the instructor and TA may examine your code, and suspicious submissions will be reported to the department if necessary. **We do not tolerate academic dishonesty**, so please obey this policy strictly.
- You are entirely responsible for making your submission in time. Back up your work **periodically**, so as to minimize the damage should any sort of computer failures occur. Follow this tutorial series on setting up a **private** Github repository for your Java projects.
- The deadline is **strict** with no excuses: you receive **0** for not making your electronic submission in time. Emailing your solutions to the instruction or TAs will not be acceptable.
- You are free to work on this lab on your own machine(s), but you are responsible for testing your code at a Prims lab machine before the submission.

Contents

Consider the following Java program (where $n = a.length$, and we assume that $n \geq 5$):

```
1  void duplicatePrint(int[] a, int n) {
2    for (int i = 0; i < n; i++) {
3      for (int j = 0; j < i; j++) {
4        for (int k = 0; k < 5; k++) {
5          System.out.println(a[k]);
6        }
7      }
8    }
9  }
```

Determine the **most accurate** asymptotic upper bound of the above program, using the big-Oh notation.

You **must** show in detail how you derive your answers. Without a convincing derivation process, you will **not** receive partial marks.

2 Submission

- **Only** computer-typed answers are acceptable. Organize your answer well: messy answers will be rejected and receive 0.
- Export your answer sheets into a PDF file: `EECS2030_F19_Lab8.pdf`
- Submit for students NOT working in a group

  If you are not working in a group, submit your solution using the submit command.

```
submit 2030 lab8 EECS2030_F19_Lab8.pdf
```

- Submit for students working in a group

  If you are working in a group, create a plain text file named `group.txt`. You can do this in eclipse using the menu `File`, then `New`, then `File`. Type your login names into the file with each login name on its own line. For example, if the students with login names rey, finn, and dameronp, worked in a group the contents of `group.txt` would be:

```
rey
finn
dameronp
```

  Submit your solution using the submit command.

```
submit 2030 lab8 EECS2030_F19_Lab8.pdf group.txt
```

- Submit your work from outside the Prism lab

  It is possible to submit work from outside the Prism lab, but the process is not trivial; do not attempt to do so at the last minute if the process is new to you. The process for submitting from outside of the Prism lab involves the following steps:

  1. transfer the files from your computer to the undergraduate EECS server red.eecs.yorku.ca
  2. remotely log in to your EECS account
  3. submit your newly transferred files in your remote login session using the instructions for submitting from within the lab
  4. repeat Steps 1 and 3 as required

  Windows users will likely need to install additional software first. Mac users have all of the required software as part of MacOS. See detailed instructions are here on how to set up remote submissions.