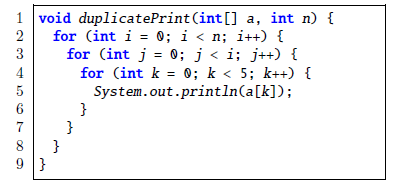
EECS2030 Fall 2019

Lab 8

Analyzing the Time Complexity of a Program

Siddarth Kanna Kannapiran

216469850



Examples:

**If n = 5**:

* The outer loop runs n times, which is 5 times.
* The middle loop runs i times, which is also 5 times.
* The inner loop runs 5 times regardless of the value of n.

Total of 125 times

**If n = 6:**

* The outer loop runs n times, which is 6 times.
* The middle loop runs i times, which is also 6 times.
* The inner loop runs 5 times regardless of the value of n.

Total of 180 times

**If n = 7:**

* The outer loop runs n times, which is 7 times.
* The middle loop runs i times, which is also 7 times.
* The inner loop runs 5 times regardless of the value of n.

Total of 245 times

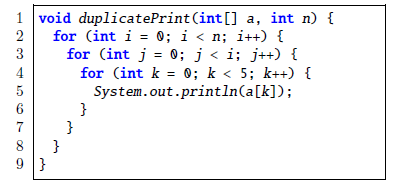
**If n = 8:**

* The outer loop runs n times, which is 8 times.
* The middle loop runs i times, which is also 8 times.
* The inner loop runs 5 times regardless of the value of n.

Total of 320 times

All the above examples run for time.

The inner most loop runs 5 times regardless of the value of n, so it runs a constant time.

The number of times the other two loops runs will be dependent on the value of n.

**Constant time**

**O (n)**

**O (5)**

**O (n)**

We have two loops that run n times, and one that runs 5 times. If we multiply all of them together, we get the expression **O (5n2)** which is the same as **O (n2)**. Therefore, this program has an asymptotic upper bound of **O (n2).**