Eecs2030- LAB 8

Consider the following Java program (where n = a:length, and we assume that n _ 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.

Solution

This is O(n^2) because even though we got there nested loops, last loop is O(1) which is in fourth line and for number of iterations of k. For first loop which is on second line, pattern for i is 0,1,2,…n-1. For every iteration of every possible value of i, we have to worry about pattern for j as well. When i is 0 and second loop doesn't run. When i is 1 j can only have the value of "0" and the loop runs 1 time. Next, if (i = 2), then j can have the values of "0 and 1" and the loop runs 2 times. Next, if (i = 3), then j can have the values of "0 and 1 and 2" and the loop runs 3 times. It will continue until the last (i = n-1), and j can have all the values "from 0 until (n-2)" and the loop runs (n-1) times. Now, we have to look at line # 4 and #5 since we have to see what is going to be running time for body of inner loop. Outer loop which is on line #4 is just O(1) since this loop will run only 5 times because variable i will start from 0 until it reaches 5. So, this part will run 5 times for each possible values of j. In the body of this loop we have a “print line” that is O(1). For example, when (i = 2) then j have possible values of “0 and 1” and then when (j = 0) third loop runs 5 times, and when (j = 1), third loop runs 5 times. In conclusion for the third loop, we can say that this loop is O(1).

In conclusion,

For (i = 0), program doesn’t run.

For (i = 1), we have (j = 0), and program runs 1 time. Third loop runs 5 times each and it is O(1).

For (i = 2), we have (j = 0) and (j = 1), and program runs 2 times. Third loop runs 5 times each and it is O(1).

For (i = 3), we have (j = 0) and (j = 1) and (j = 2), and program runs 3 times. Third loop runs 5 times each and it is O(1).

It continues ... until for (i = n-1), we have (j = 0) until (j = n-2), and program runs n-1 times. Third loop runs 5 times each and it is O(1).

We can calculate its most accurate asymptotic upper bound by adding these number of iterations.

$$O(1 + 2 + 3 + ... + (n-1)) * O(1) = O(n(n-1)/2) * O(1) = O(n^2) *O(1) = O(n^2)$$

Most accurate asymptotic upper bound of this program is O(n^2)