

## Assignment 6 Part 2.1 – Time

Ryan Brinson

10/17/2023

Output:

```
rbrinson2@DESKTOP-U8KJ40P:~/Documents/CS3305/Assignments/A6/A6P2$ java A6P21.java

Begin Time Complexity test
Current time: 22
Current time: 174
Current time: 1505
Current time: 14704

      T(n) = O(n)
```

Code:

```
// Name: Ryan Brinson
// Class: CS 3305/W04
// Term: Fall 2023
// Instructor: Carla McManus
// Assignment: 06-Part-2.1-Time

public class A6P21 {
    public static void main(String[] args) {

        // Call the function that tests time complexity
        bigOFunc();

    }

    // ----- Methods -----

    // Function that converts miles to kilometers
    public static Double milesToKilometers(Integer mi) {
        return (mi * 1.6);
    }

    // Function that shows time complexity
    public static void bigOFunc(){

        // Record the initial time for future relative time calculation
```

```
long startTime = System.currentTimeMillis();
long currTime;

System.out.println("\nBegin Time Complexity test");

// First Cycle
// Iterates 10^6 times
for (int i = 0; i < Math.pow(10, 6); i++) {
    milesToKilometers(10);
}
// Get the current time
currTime = System.currentTimeMillis();
// Print the difference from the start to the current
// to get relative time
System.out.println("Current time: " + (currTime - startTime));

// Second Cycle
// Iterates 10^7 times
for (int i = 0; i < Math.pow(10, 7); i++) {
    milesToKilometers(10);
}

// Get the current time
currTime = System.currentTimeMillis();
// Print the difference from the start to the current
// to get relative time
System.out.println("Current time: " + (currTime - startTime));

// Third Cycle
// Iterates 10^8 times
for (int i = 0; i < Math.pow(10, 8); i++) {
    milesToKilometers(10);
}

// Get the current time
currTime = System.currentTimeMillis();
// Print the difference from the start to the current
// to get relative time
System.out.println("Current time: " + (currTime - startTime));

// Fourth and final cycle
// Iterates 10^9 times
for (int i = 0; i < Math.pow(10, 9); i++) {
    milesToKilometers(10);
}
```

```
}

// Get the current time
currTime = System.currentTimeMillis();
// Print the difference from the start to the current
// to get relative time
System.out.println("Current time: " + (currTime - startTime));

// Because the cycles are linear in nature
// we end up with a linear time complexity
System.out.println("\n\t T(n) = O(n)");
}
}
```