

## Lab 3

### Part-1

Understanding runtime complexity. This lab will provide you some practice on understanding the runtime complexity of programs. The code is provided below.

**Note: No need to run the code.**

Q1. Write down complexity in terms of Big-O for methodA, method and methodC

```
import java.util.Date;
```

```
import java.util.Timer;
```

```
public class ComplexityExperiment {  
    public static void main (String args []) throws InterruptedException {  
        run_method_A(250);  
        run_method_A(500);  
        run_method_A(1000);  
        run_method_A(2000);  
        System.out.println();  
  
        run_method_B(250);  
        run_method_B(500);  
        run_method_B(1000);  
        run_method_B(2000);  
        System.out.println();  
  
        run_method_C(250);  
        run_method_C(500);  
        run_method_C(1000);  
        run_method_C(2000);  
        System.out.println();  
    }  
}
```

```
Public static void run_method_A(int n) {  
    int i = 0;  
    double start, end;
```

```

        start = System.currentTimeMillis();
        methodA(n);
        end = System.currentTimeMillis() – start;
        System.out.println("methodA(n=" + n + ") time = "+ end + "ms");
    }

```

```

public static void run_method_B(int n){
    int i = 0, loop = 1000;
    double start, end;
    start = System.currentTimeMillis();
    for(i = 0; i < loop; i++) {
        methodB(n);
    }
    end = System.currentTimeMillis() – start;
    System.out.println("methodE(n = " + n + ") time = "+ end/loop + "ms");
}

```

```

public static void run_method_C(int n){
    int i = 0;
    double start, end;
    start = System.currentTimeMillis();
    methodC(n);
    end = System.currentTimeMillis() – start;
    System.out.println("methodC(n = " + n + ")time = " + end + "ms");
}

```

```

public static void methodA(int n){

    int i = 0;
    int j = 0;
    int k = 0;
    int total = 0;
    while (i<n){
        while (j<n){
            while (k<n){

```

```

        total++;
        k++;
    }
    k=0;
    j++;
}
j=0;
i++;
}
}

```

```

Public static void method(int n){
    int i = 0;
    int j = 0;
    int total = 0;
    while (i<n){
        while (j<n){
            total++;
            j++;
        }
        i++;
    }
}

```

```

public static void methodC(int n){
    Int i = 0;
    Int j = 0;
    Int total = 0;

    j = n;
    While ((j = j/2) > 0){
        for(i = 0; i < 100*n; i++)
            total++;
    }
}

```

```
}
```

```
}
```

## Part-2

### Q2. You need to implement a two-dimensional array

Create a program that instantiates a 5x5 two-dimensional array of integers, fill it with random integers come from the range of 1 to 100. There are many ways to get random integer from 0-100, for example: `(int)(Math.random()*100)`

Outputs:

The biggest number in the array and the index of the biggest number.

The index of the row with the highest sum among all the rows and the index of the column with the highest sum among all the columns.

Output can be in the following format

```
$java two_dimensional_array
```

```
60    3    44    53    7
80    35    68    41    83
81    71    57    51    96
46    79    54    27    0
59    33    89    32    14
```

The biggest number in the array is 96, and it is index is [2][4]

The index of the row with the highest sum among all the rows is 2, and the index of the column with the highest sum among all the columns is 0

Time taken to execute the program: XX.XX ms

Also timestamp your program and record time.

In the above code, there is timestamp recorded.

Please see below code for timestamp.

```
double start,end;
```

```
start = System.currentTimeMillis();
```

```
methodA(n);
```

```
end = System.currentTimeMillis() – start;
```

You can put timestamp before and after your program.

For timestamp you may get “0ms” answer. Try for nanoseconds and see.

### What to turn in?

For local students and if possible Internet section students, demonstrate your program during lab or office hour and submit a final copy of your source code on blackboard before the assigned due date. For students in India, submit a copy of your source code on blackboard before the assigned due date.