

# Data Structures

## Exercise Booklet 2: Basic Complexity

### Exercise 1

Determine the time growth rate of the following code. You must provide details on how it was established. You may assume that  $n > 1$ .

```
for(i=1; i<n; i++) {  
    for(j=1; j<n; j*=2) {  
        System.out.println(i + " " + j);  
    }  
}
```

### Exercise 2

Determine the time growth rate of the following code. You must provide details on how it was established. You may assume that  $n > 1$ .

```
for(i=1; i<n; i++) {  
    for(j=1; j<n; j*=2) {  
        break;  
    }  
}
```

### Exercise 3

Determine the time growth rate of the following code. You must provide details on how it was established. You may assume that  $n > 1$ .

```
for(i=n-1; i>=0; i--) {  
    for(j=n-1; j>i; j--) {  
        System.out.println(i + " " + j);  
    }  
}
```

### Exercise 4

Determine the time growth rate of the following code. You must provide details on how it was established. You may assume that  $n > 1$ .

```
for(i=n-1; i>=0; i--) {  
    for(j=9; j>0; j--) {  
        System.out.println("hello");  
    }  
}
```

### Exercise 5

Consider the following snippet of code

```
for(i=0; i<n; i++) {  
    ...  
}
```

where the missing line is not provided to you. Can you assert that this code will run in  $\mathcal{O}(n)$  independently of the missing line of code? If your answer is no, then provide a counterexample.

### Exercise 6

Let  $f(n) = 100n^4 + 5000n + 3$ . Is  $f(n) \in \mathcal{O}(n^4)$ ? If yes, then justify your answer by supplying the appropriate positive constants  $c$  and  $n_0$ .

### Exercise 7

Determine the time growth rate of the following code. You must provide details on how it was established. You may assume that  $n > 1$ .

```
for (i=0; i<n; i++) {
    if (i%2==0) {
        for (j=0; j<n; j++) {
            System.out.println("Hi");
        }
    }
}
```

### Exercise 8

Determine the time growth rate of the following code. You must provide details on how it was established. You may assume that  $n > 1$ .

```
for (i=0; i<n; i++) {
    if (i%2==0) {
        for (j=0; j<i; j++) {
            System.out.println("Hi");
        }
    }
}
```

### Exercise 9

Consider the code below. Indicate:

1. How many times it prints a message.
2. Its complexity.

You may assume that  $n > 1$ .

```
for (i=0; i<n; i++) {
    for (j=0; j<n; j++) {
        if (i%2==0) {
            System.out.println("Hi");
        }
    }
}
```

### Exercise 10

Provide an example of code that has time growth rate of  $\mathcal{O}(n \log n)$ .