

## Competency #4

### Time Complexity:

Question 2:

- The correct answer would be  $O(N \lg N)$  because the first method equals to  $N$ , for the second loop, the increments goes up by multiplication of two, so logarithmic. Results in  $N * \logarithmic * 1. \Rightarrow O(N \lg N)$

### Question 2

What is the order of complexity of this segment of code?

```
for (int j = 1; j < N; j++)  
    for (int k = 1; k < N; k=k*2)  
        System.out.println (j + k);
```

- ☐  $O(2^N)$
- ☐  $O(\lg N)$
- ☐  $O(N^2)$
- ☐  $O(N \lg N)$
- ☐  $O(1)$
- ☐  $O(N)$
- ☒ None of the above

### Recognizing common growth functions:

2 constant

5 linear

4 log

- The constant one should be #2
- The linear one should be #5
- The logarithmic one should be #4
-