

1 What's Faster?

For each example below, there are two algorithms solving the same problem. Given the asymptotic runtimes for each, is one of the algorithms **guaranteed** to be faster? If so, which? And if neither is always faster, explain why. Assume the algorithms have very large input (i.e. N is very large).

- (a) Algorithm 1: $\Theta(N)$, Algorithm 2: $\Theta(N^2)$

- (b) Algorithm 1: $\Omega(N)$, Algorithm 2: $\Omega(N^2)$

- (c) Algorithm 1: $O(N)$, Algorithm 2: $O(N^2)$

- (d) Algorithm 1: $\Theta(N^2)$, Algorithm 2: $O(\log N)$

- (e) Algorithm 1: $O(N \log N)$, Algorithm 2: $\Omega(N \log N)$

Why do we need to assume that N is large?

2 Basic Algorithmic Analysis

For each of the following function pairs f and g , list out the Θ, Ω, O relationships between f and g , if any such relationship exists. For example, $f(x) \in O(g(x))$.

(a) $f(x) = x^2, g(x) = x^2 + x$

(b) $f(x) = 50000x^3, g(x) = x^5$

(c) $f(x) = \log(x), g(x) = 5x$

(d) $f(x) = e^x, g(x) = x^5$

(e) $f(x) = \log(5^x), g(x) = x$

3 Practice with Runtime

For each of the following functions, find the Big-Theta expression for the runtime of the function in terms of the input variable n .

You may find the following relations helpful:

$$1 + 2 + 3 + 4 + \dots + N = \Theta(N^2)$$

$$1 + 2 + 4 + 8 + \dots + N = \Theta(N)$$

- (a) For this problem, you may assume that the static method `constant` runs in $\Theta(1)$ time.

```

1  public static void bars(int n) {
2      for (int i = 0; i < n; i += 1) {
3          for (int j = 0; j < i; j += 1) {
4              System.out.println(i + j);
5          }
6      }
7
8      for (int k = 0; k < n; k += 1) {
9          constant(k);
10     }
11 }
```

- (b) Determine the runtime for `barsRearranged`.

```

1  public static void cowsGo(int n) {
2      for (int i = 0; i < 100; i += 1) {
3          for (int j = 0; j < i; j += 1) {
4              for (int k = 0; k < j; k += 1) {
5                  System.out.println("moove");
6              }
7          }
8      }
9  }
10
11 public static void barsRearranged(int n) {
12     for (int i = 1; i <= n; i *= 2) {
13         for (int j = 0; j < i; j += 1) {
14             cowsGo(j);
15         }
16     }
17 }
```