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 $\Theta, \Omega, O$ 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:

\[
1 + 2 + 3 + 4 + \cdots + N = \Theta(N^2)
\]

\[
1 + 2 + 4 + 8 + \cdots + N = \Theta(N)
\]

(a) For this problem, you may assume that the static method \texttt{constant} runs in \( \Theta(1) \) time.

```java
public static void bars(int n) {
    for (int i = 0; i < n; i += 1) {
        for (int j = 0; j < i; j += 1) {
            System.out.println(i + j);
        }
    }
    for (int k = 0; k < n; k += 1) {
        constant(k);
    }
}
```

(b) Determine the runtime for \texttt{barsRearranged}.

```java
public static void cowsGo(int n) {
    for (int i = 0; i < 100; i += 1) {
        for (int j = 0; j < i; j += 1) {
            for (int k = 0; k < j; k += 1) {
                System.out.println("moove");
            }
        }
    }
}
```

```java
public static void barsRearranged(int n) {
    for (int i = 1; i <= n; i *= 2) {
        for (int j = 0; j < i; j += 1) {
            cowsGo(j);
        }
    }
}
```