1. Assume you have a complex multiplication where the terms have very large digits. What will be the speedup from using Gauß multiplication?

\[
\text{speedup} = \frac{\text{naive runtime}}{\text{fast runtime}}. \quad \text{Asymptotically, additions and subtractions are negligible compared to multiplications; therefore, the speedup will be \(\frac{4}{3}\). It is only a constant-time speedup, \textit{i.e.}, not a significant algorithmic speedup.}
\]

\textbf{Short answer:} 4/3

2. Assume you have an integer multiplication on two integers with many digits. What will be the speedup from using Karatsuba multiplication?

\[
\text{speedup} = \frac{\Theta(n^2)}{\Theta(n^{1.585})}; \quad \text{this will grow in an unbounded manner as } n \text{ gets large. Thus, the speedup becomes infinite when } n \text{ becomes large.}
\]

\textbf{Short answer:} infinite

3. Prove the runtime of Karatsuba multiplication using the master theorem.

\[
r(n) = 3r(n/2) + \Theta(n^1)
\]

\[
log_2(3) \approx 1.585
\]

\[
> 1; \quad \text{we compare to 1, because it’s the power of } n^1
\]

\[
\rightarrow \text{leaf-heavy.}
\]

\[
r(n) \in \Theta(n^{\log_2(3)}).
\]
4. Let \( x = 1234567890 \) and \( y = 9876543210 \). What are \( x_{\text{high}} \), \( x_{\text{low}} \), \( y_{\text{high}} \), and \( y_{\text{low}} \)?

\[
\begin{align*}
  x_{\text{high}} &= 12345 \\
  x_{\text{low}} &= 67890 \\
  y_{\text{high}} &= 98765 \\
  y_{\text{low}} &= 43210
\end{align*}
\]

5. In the question above, assume that you are given the following from recursive calls:

\[
\begin{align*}
  z_{\text{high}} &= x_{\text{high}} \cdot y_{\text{high}} = 1219253925 \\
  z_{\text{low}} &= x_{\text{low}} \cdot y_{\text{low}} = 2933526900 \\
  z_{\text{total}} &= (x_{\text{high}} + x_{\text{low}}) \cdot (y_{\text{high}} + y_{\text{low}}) = 11391364125
\end{align*}
\]

Find \( z_{\text{medium}} \) using Karatsuba’s method. Then find \( z = x \cdot y \). Verify this against the value of \( z = x \cdot y \) computed using Python’s built-in integers.

\[
\begin{align*}
  z_{\text{total}} &= (x_{\text{high}} + x_{\text{low}}) \cdot (y_{\text{high}} + y_{\text{low}}) \\
  &= z_{\text{high}} + z_{\text{medium}} + z_{\text{low}} \\
  z_{\text{medium}} &= z_{\text{total}} - z_{\text{high}} - z_{\text{low}} \\
  &= 7238583300.
\end{align*}
\]

\[
\begin{align*}
  z &= z_{\text{high}} \cdot 10^n + z_{\text{medium}} \cdot 10^5 + z_{\text{low}} \\
  &= z_{\text{high}} \cdot 10^{10} + z_{\text{medium}} \cdot 10^5 + z_{\text{low}} \\
  &= 12193263111263526900.
\end{align*}
\]

Using the naive approach, we get \( x \cdot y = 1234567890 \cdot 9876543210 = 12193263111263526900 \), which matches.