1. Rank the following functions by their asymptotic growth rate in ascending order. In your solution, group those functions that are big-Theta of one another (all log functions are base 2):

\[
\begin{array}{cccccc}
6 \cdot n \log n & 2^{100} & \log \log n & \log^2 n & 2^{\log n} \\
2^n & \lfloor \sqrt{n} \rfloor & n^{0.01} & 1/n & 4n^{3/2} \\
3n^{0.5} & 5n & [2 \cdot n \log^2 n] & 2^n & n^2 \\
4^n & n^3 & n^2 \log n & 4^{\log n} & \sqrt{\log n} \\
\end{array}
\]

2. Algorithm algo1 uses \(10n \log n\) operations, while algorithm algo2 uses \(n^2\) operations. What is the value of \(n_0\), such that algo1 is better than algo2 for \(n \geq n_0\).

3. For each of the following, give a big-Oh characterization in terms of \(n\) (count additions and multiplications):

(a) \(s = 1\)
for \(i = 1\) to \(2n\) do
\(s = s \cdot i\)

(b) \(s = 0\)
for \(i = 1\) to \(2n\) do
for \(j = 1\) to \(i\) do
\(s = s + i\)

(c) \(s = 0\)
for \(i = 1\) to \(n\cdot n\) do
for \(j = 1\) to \(i\) do
\(s = s + i\)

4. Suppose you are given an algorithm A to find an element \(e\) in an \(n \times n\) matrix \(M\). The algorithm A iterates over the rows of \(M\) and calls another algorithm B on each one, until \(e\) is found or it has searched all rows of \(M\). What is the worst-case running time of A in terms of \(n\)? (B performs a linear search in \(O(n)\) time)
5. Suppose you run two algorithms, P and Q, on many randomly generated data sets. P is an \( O(n \log n) \)-time algorithm and Q is an \( O(n^2) \)-time algorithm. After your experiments you find that if \( n < 100 \), Q actually runs faster, and only when \( n \geq 100 \), P is faster. Explain why this scenario is possible, including numerical examples.

6. An array \( A \) contains \( n - 1 \) unique integers in the range \([0, n - 1]\); that is, there is one number from this range that is not in \( A \). Describe an \( O(n) \) time algorithm for finding that number. You are allowed to use only \( O(1) \) additional memory besides the array \( A \) itself.

7. Suppose that each row of an \( n \times n \) matrix \( M \) consists of only binary digits, such that, in any row of \( M \), all the 1’s come before any 0’s in that row. Assuming \( M \) is already in memory, describe an \( O(n) \)-time algorithm for finding the row of \( M \) that contains the most 1’s.

8. Given an array \( A \) of \( n - 2 \) unique integers in the range \([1, n]\), describe an \( O(n) \)-time algorithm for finding the two integers in the range \([1, n]\) that are not in \( A \). You are allowed to use only \( O(1) \) additional memory besides the array \( A \) itself.

9. An \( n \)-degree polynomial \( p(x) \) is an equation of the form:

\[
p(x) = \sum_{i=0}^{n} a_i x^i
\]

where \( x \) is a real number and each \( a_i \) is a constant.

(a) Describe a simple \( O(n^2) \)-time method for computing \( p(x) \) for a particular value of \( x \).

(b) Consider now a rewriting of \( p(x) \) as:

\[
p(x) = a_0 + x(a_1 + x(a_2 + \cdots + x(a_{n-1} + xa_n)\ldots))
\]

which is known as Horner’s method. Using the big-Oh notation, characterize the number of multiplications and additions this method uses.

10. Give the values of the maximum suffix sums \( M_t \) for the array \([-2, -4, 3, -1, 5, 6, -7, -2, 4, -3, 2]\).

11. Describe how to modify the MaxsubFastest algorithm so that it uses just a single loop and, instead of computing \( n + 1 \) different \( M_t \) values, it maintains just a single variable \( M \).

12. Given an array \( A \), of \( n \) integers, describe a method to find the longest subarray of \( A \) such that all the numbers in that subarray are in sorted order. What is the running time of your algorithm?

13. Given the array \( A \) with elements \([22, 15, 36, 44, 10, 3, 9, 13, 29, 25]\), illustrate the performance of the selection-sort algorithm on \( A \).

14. Given the array \( A \) with elements \([22, 15, 36, 44, 10, 3, 9, 13, 29, 25]\), illustrate the performance of the insertion-sort algorithm on \( A \).
15. Let $S$ be an array of $n$ elements. An inversion in $S$ is a pair of indices $i$ and $j$ such that $i < j$ but $S[i] > S[j]$. Give an example of a sequence of $n$ integers with $\Omega(n^2)$ inversions.