2.) From the results obtained, we can observe the following:

For arrays of size 10000:

Insertion Sort took approximately 12.08 milliseconds.
Selection Sort took approximately 20.33 milliseconds.
Bubble Sort took approximately 100.84 milliseconds.
Merge Sort took approximately 0.93 milliseconds.
Quick Sort took approximately 0.65 milliseconds.

For arrays of size 100:

Insertion Sort took approximately 0.0028 milliseconds.
Selection Sort took approximately 0.0069 milliseconds.
Bubble Sort took approximately 0.0104 milliseconds.
Merge Sort took approximately 0.0034 milliseconds.
Quick Sort took approximately 0.0034 milliseconds.
From these results, we can observe that:

- Merge Sort and Quick Sort are significantly faster for both array sizes than Insertion Sort, Selection Sort, and Bubble Sort.

- As expected, the sorting algorithms with O(n^2) time complexity (Insertion Sort, Selection Sort, Bubble Sort) take significantly longer for larger array sizes compared to the
- O(n log n) time complexity algorithms (Merge Sort, Quick Sort).

- Merge and Quick Sort show consistent performance across different array sizes, indicating their scalability.

- Insertion Sort, Selection Sort, and Bubble Sort exhibit quadratic time complexity behavior, leading to significant increases in execution time as the array size grows.

3.)
Insertion Sort:

Estimated Runtime ≈ (12.08 / 10000^2) milliseconds * (1,000,000,000)^2 ≈ 1.208 * 10^16 milliseconds

Selection Sort:

Estimated Runtime ≈ (20.33 / 10000^2) milliseconds * (1,000,000,000)^2 ≈ 2.033 * 10^16 milliseconds

Bubble Sort:

Estimated Runtime ≈ (100.84 / 10000^2) milliseconds * (1,000,000,000)^2 ≈ 1.0084 * 10^18 milliseconds

Merge Sort:

Estimated Runtime ≈ (0.93 / (1000 * log(1000))) milliseconds * (1,000,000,000) * log(1,000,000,000) ≈ 3.724 * 10^9 milliseconds

Quick Sort:

Estimated Runtime ≈ (0.65 / (1000 * log(1000))) milliseconds * (1,000,000,000) * log(1,000,000,000) ≈ 2.6 * 10^9 milliseconds

These are the estimated runtimes for sorting an array of size one billion (1,000,000,000) using the estimated values of c and the asymptotic formulas.