**3. Analyze the time complexity of the Quicksort algorithm:**

The time complexity of the average case for quicksort is O(n log n), where n is the number of elements in the array.

**4. Compare Quicksort with Merge Sort:**

Both quicksort and merge sort have an average time complexity of O(n log n). However, quicksort is generally more efficient in practice due to its smaller constant factors.

**5. Discuss the advantages and disadvantages of Quicksort:**

**Advantages**:

Efficient on average, especially for large datasets.

In-place sorting, requiring only a constant amount of additional memory.

**Disadvantages**:

Worst-case time complexity is O(n^2), which occurs when the pivot is always the smallest or largest element.

Not stable (order of equal elements might change).

**6. Write a conclusion that summarizes the strengths and weaknesses of Quicksort and its performance compared to other sorting algorithms.**

Quicksort is a fast and efficient sorting algorithm with an average-case time complexity of O(n log n). Its in-place nature makes it suitable for large datasets, but it has a potential disadvantage of O(n^2) in the worst case. In comparison to other sorting algorithms like merge sort, quicksort often performs better in practice due to its smaller constant factors. However, the choice of sorting algorithm depends on the specific requirements and characteristics of the data being sorted.