Misconceptions

Module-3	
Misconception 1.	**An algorithm that runs faster in practice always has a better timecomplexity.**
Correct Explanation	The concept of time complexity offers a comprehensive comprehension of how the performance of an algorithm is influenced by the amount of the input. In practical scenarios, it is possible for an algorithm with a greater theoretical time complexity to exhibit shorter execution times for lower inputsizes or as a result of optimisation techniques used at the implementation level.
Misconception 2.	**Big-O notation gives the exact running time of an algorithm.**
Correct Explanation	The Big-O notation is a mathematical language used to express an asymptotic upper limit on the growth rate of an algorithm's running time. It does not offer a precise measurement of the number of operations executed by the algorithm. This method facilitates the comparison of algorithm scalability, albeit it does not take into consideration constants or lower-order terms.

Misconception 3.	**Recursive algorithms often exhibit increased temporal complexitycompared to their iterative counterparts.**
Correct Explanation	The time complexity of an algorithm is not fundamentally determined by whether it is recursive or iterative. The efficacy of an algorithm is contingent upon its logical and structural components, rather than only relying on the useof recursion.
Misconception 4.	**Time complexity takes precedence over space complexity in most cases.**
Correct Explanation	The consideration of time complexity is of utmost significance in computational analysis; however, it is imperative to acknowledge that spacecomplexity also has great value, particularly in systems with constrained memory resources. In some contexts, such as embedded systems or large-scale data processing, the importance of optimising for space might be equivalent to optimising for time.
Misconception 5.	**The selection of an algorithm with a lower Big-O complexity isconsistently favoured.**

Correct Explanation	Although Big-O notation offers valuable information about the worst-case performance of algorithms, it is important to acknowledge that real-world performance may deviate from these theoretical predictions owing to variables such as input distribution, system-specific characteristics, or lower-order terms. When selecting an algorithm, it is important to take intoaccount both theoretical analysis and actual testing.