
Summary

There are requirements for time complexity and space complexity for most coding interview problems, and usually interviewers pay more attention to time complexity. However, candidates can always ask interviewers about their requirements when in doubt.

The first strategy to improve time efficiency is to use appropriate data structures. Different types of data structures are suitable for different scenarios. Candidates should consider the pros and cons for each type of data structures and make the most appropriate choice.

The second strategy is to apply appropriate algorithms. For example, the binary search algorithm accelerates searches in a sorted array, and dynamic programming algorithms make it simpler to find the optimized solutions (minimum or maximum values) for many problems. Sometimes mathematical proofs are necessary to demonstrate the correctness of algorithms.

In many cases, we have to sacrifice memory to improve time efficiency. A lookup table (an array or a 2D matrix in most cases) can be utilized to avoid recalculations, and a hash table usually facilitates character searches in a string.