

Summarize what time and space complexity means and explain how you get it.

---

Time complexity estimates how the computation time of an algorithm scales with the growth of its input.

There are several methods for determining the time complexity of an algorithm:

1. Counting the number of basic operations (assignments, comparisons, arithmetic operations, and accessing data in arrays) performed by the algorithm as a function of the input size
2. Big O Notation represents the worst-case scenario for the growth rate of the algorithm's running time
3. For recursion, we set up a recurrence relation that expresses the time complexity in terms in smaller subproblems
4. We can also use asymptotic analysis that focuses on the behavior of the algorithm as the input size approaches infinity

Space complexity estimates how the memory (arrays, variables) used by the algorithm scales with the growth of its input.

There are several methods for determining the time complexity of an algorithm:

1. Count the amount of memory used by the algorithm so we can keep track of the variables and data structures that grow with the input size
2. Using Big O Notation, we can express the upper bound of the algorithm's space complexity in terms of the input size
3. For algorithms that involve recursion or dynamic programming, we also set up a recurrence relation that expresses the space complexity in terms of the input size and smaller subproblems
4. Using asymptotic analysis, we can analyze the behavior of the algorithm's memory usage as the input size approaches infinity