Complexity of algorithms and big O notation

* Big O Notation is a tool used to describe the time complexity of algorithms. It calculates the time taken to run an algorithm as the input grows. In other words, it calculates the worst-case time complexity of an algorithm.
* Algorithms may involve loops and conditionals impacting running time. Efficiency matters, especially with large inputs.
* Computational complexity is defined as how long an algorithm takes concerning input size. The lower the complexity, the more efficient the algorithm. Focus on order of magnitude, especially for the longest loop.
* Big O notation expresses the worst possible runtime. Examples: 𝒪(n) grows proportionally to input, 𝒪(n^2) grows with the square of input. Comparison is based on functions and their asymptotic behavior.
* Big O helps compare running time of different functions. Not relying on direct time measurement; instead, comparing growth rates. Graphs show different complexities: linear (𝒪(x)), quadratic (𝒪(x^2)), etc.
* Function complexity based on behavior as input approaches infinity. Not all 'linear' functions are strictly linear in computer programs. Introducing the concept of 𝒪(g(x)) with witnesses C and k.
* In computer science, Big O Notation is a fundamental tool used to find out the time complexity of algorithms.

Calculating witness pairs

* Witnesses C and k needed to express f(x) = 𝒪(g(x)).
* Start with a value for k and find C such that for x ≥ k, |f(x)| ≤ Cg(x). Example: 𝒪(x²) with k = 1, C = 4; k = 2, C = 3.
* For a function f(x) = anxn+an−1xn−1+…+a0, where ≠0an=0, f(x) is 𝒪(x^n).
* Choosing k = 1 often a safe first choice.
* Big O hierarchy to compare functions from lowest to highest order.
* Combining inequalities to find witness pairs for specific functions.
* In computer science, Big O Notation is a fundamental tool used to find out the time complexity of algorithms.

Reference

1. Simplilearn. (2023, November 6). *Introduction to Big O notation in data structure*. Simplilearn.com. <https://www.simplilearn.com/big-o-notation-in-data-structure-article#:~:text=In%20computer%20science%2C%20Big%20O,Linear%20Search%20%E2%80%93%20O(n)>