Sequential Algorithms Complexity Analysis

Paris Gavriilidis

November 2024

1 Unique Numbers Algorithm

Main Function: findUniqueNumbers

$$
\begin{aligned}
T_{total}(n) &= O(n \log n) \\
T_{insert}(n) &= O(n) \\
T_{assign}(n) &= O(n) \\
T_{sort}(n) &= O(n \log n) \\
S_{total}(n) &= O(n) \\
S_{uniqueSet} &= O(n) \\
S_{vector} &= O(n)
\end{aligned}
$$

2 Optimized Sequence Algorithm

Main Function: findMaxConsecutiveUnique

$$
\begin{aligned}
T_{total}(n) &= O(n^2) \\
T_{createSpaced}(n) &= O(n) \\
T_{collectDuplicates}(n) &= O(n) \\
T_{sortDuplicates}(n) &= O(n \log n) \\
T_{placeDuplicates}(n) &= O(n^2) \\
T_{createResult}(n) &= O(n) \\
S_{total}(n) &= O(n) \\
S_{spacedArray} &= O(n) \\
S_{duplicates} &= O(n) \\
S_{seen} &= O(n) \\
S_{result} &= O(n)
\end{aligned}
$$

3 Brute Force Algorithm

Main Function: BruteForce

$$
\begin{aligned}
T_{total}(n) &= O(n \cdot n!) \\
T_{sort}(n) &= O(n \log n) \\
T_{permutations}(n) &= O(n!) \\
T_{subsequences}(n) &= O(n^2) \\
T_{validate}(n) &= O(n) \\
S_{total}(n) &= O(n) \\
S_{bestSequence} &= O(n) \\
S_{currentSeq} &= O(n)
\end{aligned}
$$

4 Optimal Sequence Algorithm

Main Function: findOptimalSequence

$$
\begin{aligned}
T_{total}(n) &= O(n \log n) \\
T_{countFreq}(n) &= O(n) \\
T_{sort}(n) &= O(k \log k) \text{ where } k \leq n \\
T_{buildSeq}(n) &= O(n) \\
T_{merge}(n) &= O(n) \\
S_{total}(n) &= O(n) \\
S_{freq} &= O(n) \\
S_{numFreq} &= O(n) \\
S_{seq1} &= O(n) \\
S_{seq2} &= O(n) \\
S_{result} &= O(n)
\end{aligned}
$$

5 Helper Functions

isValidSequence

$$
\begin{aligned}
T(n) &= O(n) \\
S(n) &= O(1)
\end{aligned}
$$

validateSequence

$$
\begin{aligned}
T(n) &= O(n) \\
S(n) &= O(1)
\end{aligned}
$$