Program Structures and Algorithms

Spring 2024

NAME: Kartikey Vijayakumar Hebbar

NUID: 002276938

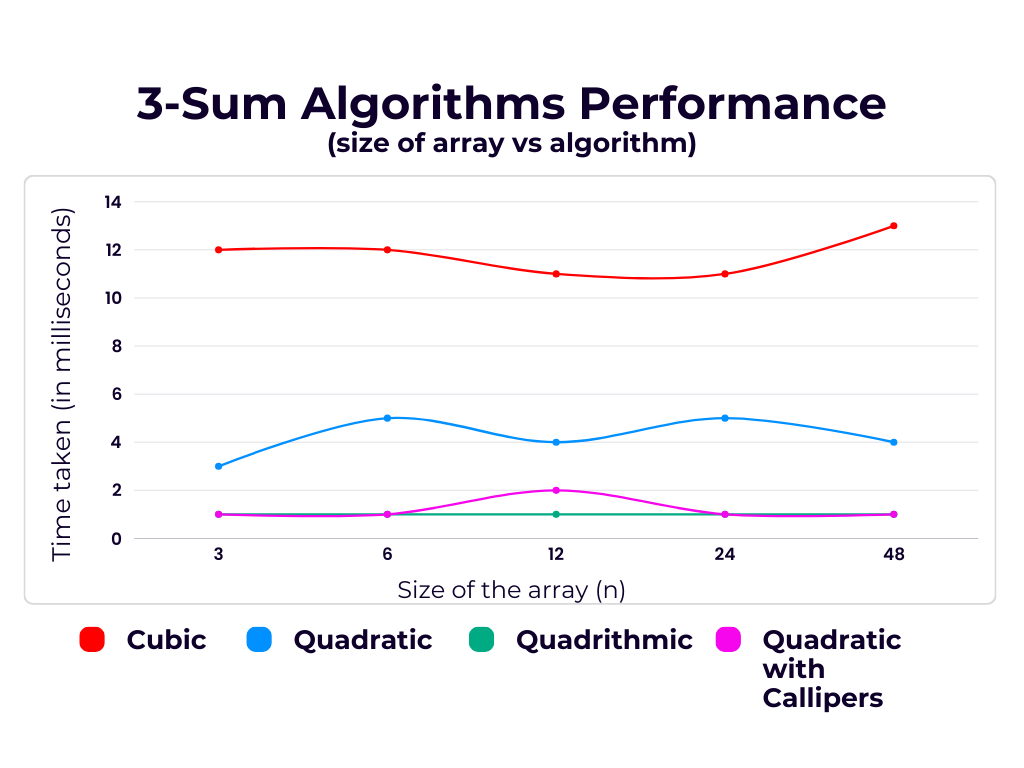
GITHUB LINK: https://github.com/kartikeyhebbar/INFO6205/tree/Spring2024/src/main/java/edu/neu/coe/info6205/threesum

**Task: Assignment 2 (3-SUM)**

**Relationship Conclusion:**

Below is a tabular comparison of how all the 4 approaches to solving this problem performed in terms of time complexity.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Size of array (n)** | **Time taken by Cubic Algorithm (in Milliseconds)** | **Time taken by Quadratic Algorithm (in Milliseconds)** | **Time taken by Quadrithmic Algorithm (in Milliseconds)** | **Time taken by Quadratic with Callipers Algorithm (in Milliseconds)** |
| 3 | 12 | 3 | 1 | 1 |
| 6 | 12 | 5 | 1 | 1 |
| 12 | 11 | 4 | 1 | 2 |
| 24 | 11 | 5 | 1 | 1 |
| 48 | 13 | 4 | 1 | 1 |



The above data shows that the Cubic approach to solving 3-Sum performed the least by taking the most time to execute while the Quadrithmic and Quadratic with Calipers approach perform way better in almost all cases of increasing array lengths.

**Evidence to support that conclusion:**

1. Poor performance of the Cubic Algorithm: This happens because in this approach, we use three nested for-loops traversing each element of the array 3 times to compute the solution. The time complexity of the program is O(n^3).
2. Better approach using Quadratic Algorithm: The quadratic algorithm solves the problem in the order of O(n^2) by making use of a HashSet to track the elements which makes it much faster than the previous approach.
3. Fastest approaches using Quadrithmic and Quadratic Calipers method: The Quadrithmic approach uses the binary search operation on the array to find the third number of the triplet which satisfies the condition, hence reduces the extra O(n) to O(logn) and makes the operation faster. The Quadratic with Calipers solution uses 2 pointer approach to reduce the need for nested for-loops. Both these algorithms solve the question in the order of O(n^2) but are more efficient in computation than the regular Quadratic solution.

The Quadratic Approaches reduce the extra overhead of O(n) by using various other techniques (like Sorting and Binary Search) to reduce the time complexity, hence they prove to be significantly faster than the Cubic approach.

**Unit Test Screenshots:**

A computer screen shot of a computer screen

Description automatically generated