**REPORT**

The issue can be formed as the need might arise to track down the upsides of the letters in the cryptarithmetic puzzle to such an extent that the condition turns out as expected. The issue can be tackled utilizing the accompanying two calculations that I have studied:

1. Depth First Search (DFS)

2. A\* algorithm

**DFS:**

The DFS algorithm is a recursive algorithm that uses the idea of backtracking. It involves an exhaustive search of all the possible combinations of the letters and their corresponding values. The algorithm starts with the first letter and assigns a value to it. It then moves to the next letter and assigns a value to it. If the value assigned to the letter results in a valid combination, the algorithm moves to the next letter. If the value assigned to the letter results in an invalid combination, the algorithm backtracks and assigns a different value to the letter. The algorithm continues this process until all the letters have been assigned values and a valid combination is found.

**A\* algorithm:**

The A\* algorithm is an informed search algorithm that uses a heuristic function to guide the search. The algorithm starts with an initial state and generates all the possible successor states. It then evaluates each successor state using the heuristic function and selects the state with the lowest evaluation value. The algorithm continues this process until a valid combination is found.

The running time complexity of the DFS algorithm is O(n!) and the space complexity is O(n), where n is the number of letters in the puzzle. The running time complexity of the A\* algorithm is O(b^d) and the space complexity is O(b^d), where b is the branching factor and d is the depth of the search

**Observations:**

The DFS algorithm is a simple algorithm that involves exhaustive search of all the possible combinations of the letters and their corresponding values. It is not an efficient algorithm and requires a lot of time and space to find a valid combination. The A\* algorithm is an informed search algorithm that uses a heuristic function to guide the search. It is a more efficient algorithm and requires less time and space to find a valid combination.

In my opinion, the A\* algorithm is the best algorithm for solving cryptarithmetic puzzles as it is more efficient and requires less time and space to find a valid combination.