Synopsis

Problem Statement

Let **A[ i…n ]** be an array of **n** distinct real numbers. A pair **( A[ i ], A[ j ] )** is said to be an index-value inversion if **A[ i ] = j** and **A[ j ] = i** . Design an algorithm for counting the number of index-value inversions.

Design Technique

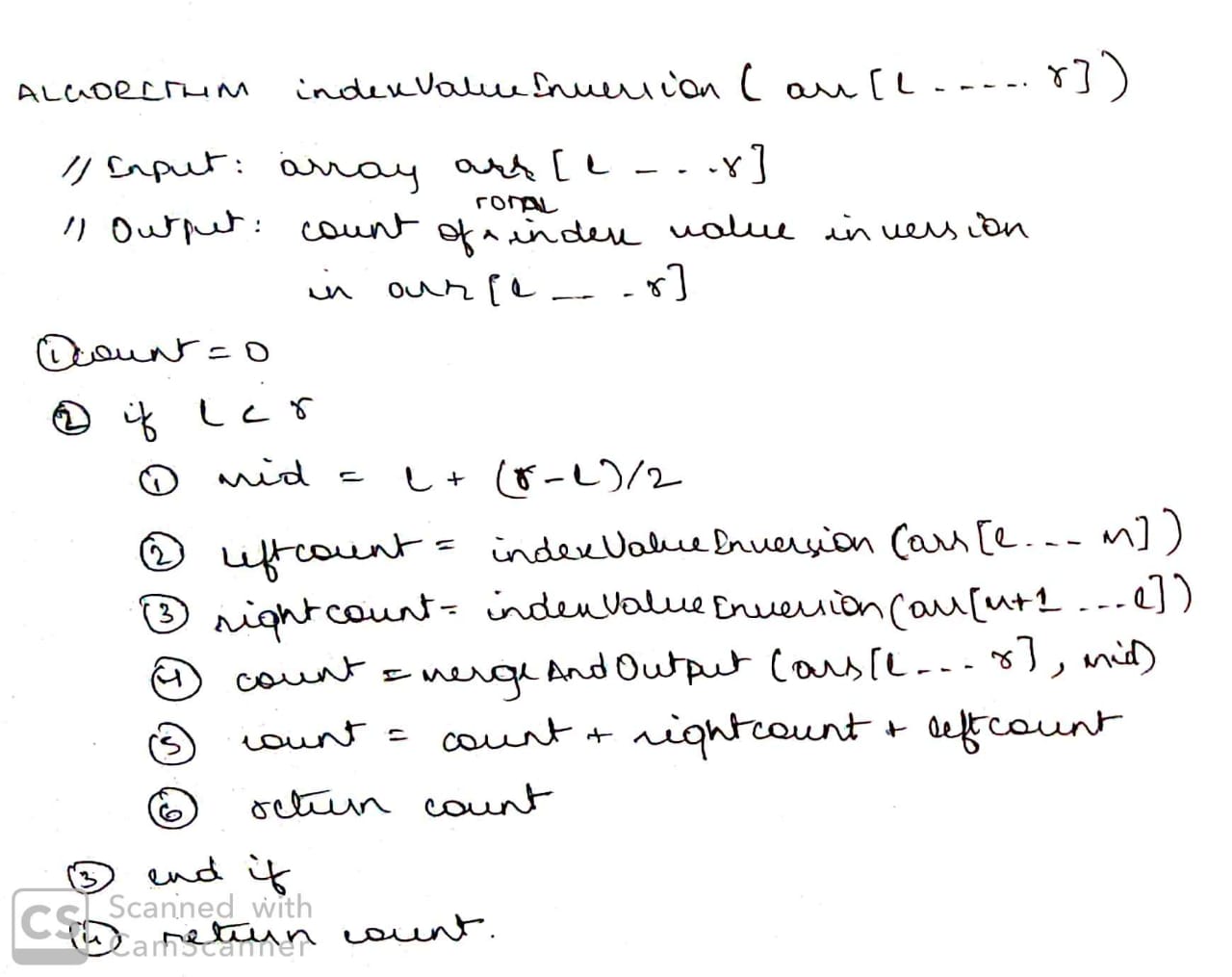
We can obtain an efficient solution for the above problem statement if we use **Divide And Conquer** **Technique**. Here, we divide the input array into two halfs. The number of **index value inversions from** **both the halves are added** to find the total number of index value inversions in the complete array. This procedure is **executed** **recursively**.

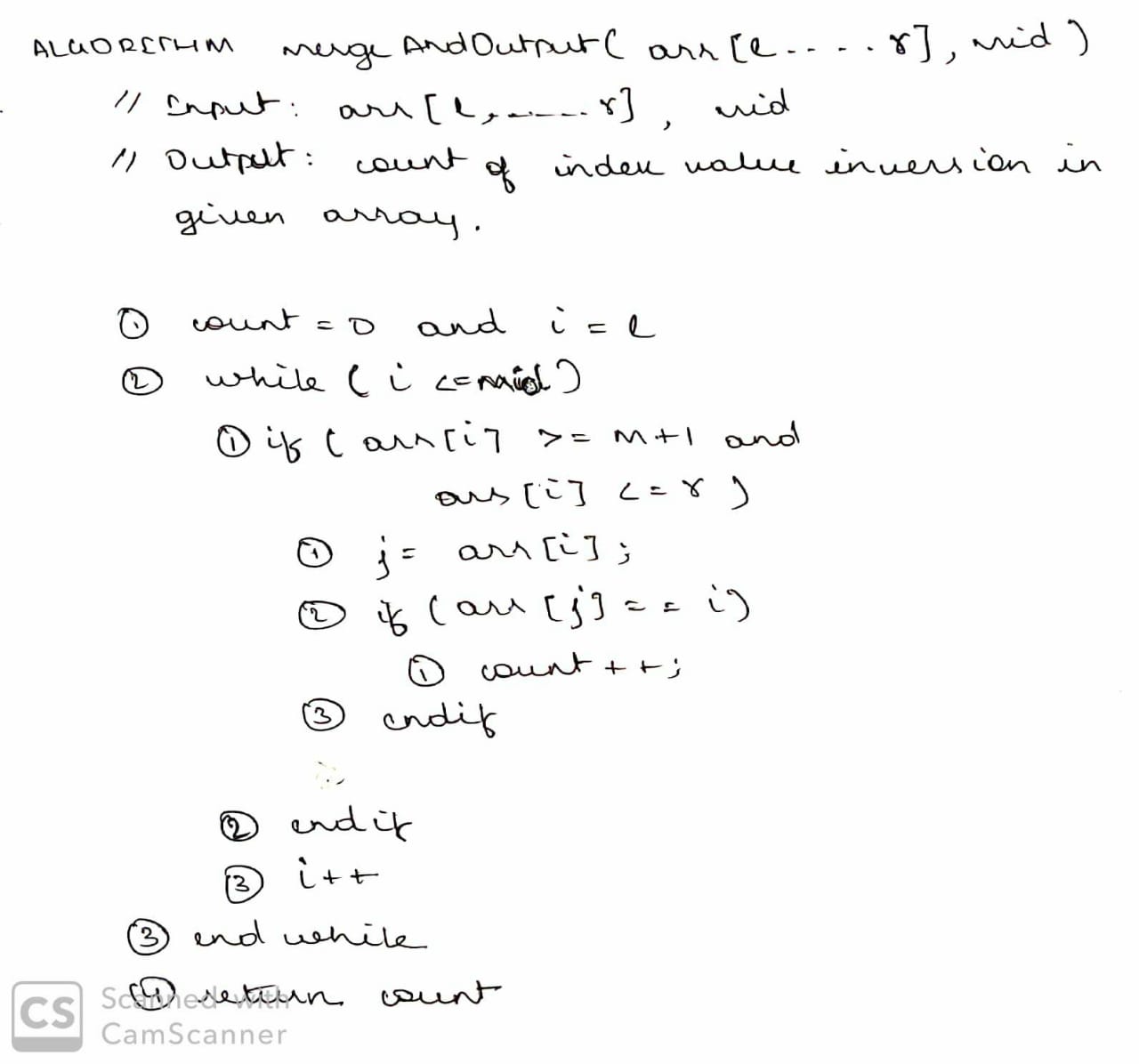
We can expect our solution to be similar to that of mergesort.

Data Structure

**Array Data Structure** is used to solve the above problem statement. We have used this data structure because :

* It is **easier to code** our algorithm around array data structure.
* It gives an **efficient solution**.
* We face **no notable loss** in time or memory efficiency.

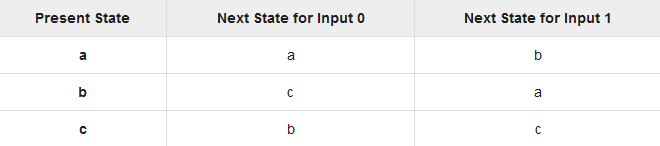
Algorithm



Algorithm Analysis

Let a deterministic finite automaton P be P = ( Q, ∑, q0, F), where Q = {a, b, c}, ∑ = {0, 1}, q0 = {a}, F = {c}.

Transition function δ as shown by the following table



Source Code

Source Code

Output

Output

References

<https://www.geeksforgeeks.org/program-to-construct-a-dfa-which-accept-the-language-l-anbm-n-mod-20-m1/>

Project Repository

<https://github.com/aksharsramesh/TOCproject>