Code Implementation:

Bubble Sort.py:

Bubble Sort works by repeatedly swapping the adjacent elements if they are in the wrong order. This algorithm is not suitable for large data sets as its average and worst-case time complexity is quite high.

A computer screen shot of a program

Description automatically generated

Time and Space Complexity:

The basic Bubble Sort has a time complexity of O(n^2) in both the average and worst cases because it always performs a fixed number of passes through the list. The optimized Bubble Sort also has a worst-case time complexity of O(n^2), but its best-case time complexity is improved to O(n) due to the early termination check when no swaps are needed. Both versions have a space complexity of O(1) since they sort the list in place without requiring additional memory.

Optimized Changes:

Bubble\_Sort\_Optimized.py is an upgraded version of Bubble Sort that uses a swapped flag to check if the array is already sorted, letting it finish faster when no more swaps are needed. This makes it quicker in cases where the list is already in order, unlike the basic Bubble Sort which always goes through the entire list.

A computer screen shot of a code

Description automatically generated

Test Cases:

* Test Case 1: Randomly generated array of integers: [64, 34, 25, 12, 22, 11, 90]
  + Outcome: Both versions correctly sorted the array to [11, 12, 22, 25, 34, 64, 90].
* Test Case 2: Already sorted array: [1, 2, 3, 4, 5, 6, 7, 8, 9]
  + Outcome: Both versions correctly returned the sorted array, but the optimized version did so more quickly due to early termination.
* Test Case 3: Array sorted in descending order: [9, 8, 7, 6, 5, 4, 3, 2, 1]
  + Outcome: Both versions correctly sorted the array to [1, 2, 3, 4, 5, 6, 7, 8, 9].
* Test Case 4: Array with all identical elements: [5, 5, 5, 5, 5]
  + Outcome: Both versions correctly returned the array unchanged as [5, 5, 5, 5, 5].
* Test Case 5: Edge case: empty array: []
  + Outcome: Both versions correctly returned the empty array unchanged.
* Test Case 6: Edge case: single-element array: [1]
  + Outcome: Both versions correctly returned the single-element array unchanged.