

144. Write code to modify bubble\_sort function to stop early if the list becomes sorted before all passes are completed.

**Test Cases:**

- Test your optimized function with the following lists:
  1. **Input:** [64, 25, 12, 22, 11]
    - **Expected Output:** [11, 12, 22, 25, 64]
  2. **Input:** [29, 10, 14, 37, 13]
    - **Expected Output:** [10, 13, 14, 29, 37]
    -

AIM: To sort an elements by using bubble sort

**PROGRAM:**

```
def bubble_sort_optimized(nums):
    n = len(nums)
    for i in range(n):
        swapped = False
        for j in range(0, n-i-1):
            if nums[j] > nums[j+1]:
                nums[j], nums[j+1] = nums[j+1], nums[j]
                swapped = True # Set swapped flag to True
        if not swapped:
            break

if __name__ == "__main__":
    arr1 = [64, 25, 12, 22, 11]
    bubble_sort_optimized(arr1)
    print("Sorted array for Test Case 1:", arr1)
    arr2 = [29, 10, 14, 37, 13]
    bubble_sort_optimized(arr2)
    print("Sorted array for Test Case 2:", arr2)
```

```
Sorted array for Test Case 1: [11, 12, 22, 25, 64]
Sorted array for Test Case 2: [10, 13, 14, 29, 37]
```

OUTPUT:

TIME COMPLEXITY:  $O(n^2)$