

132.

1. Write code to modify bubble_sort function to stop early if the list becomes sorted before all passes are completed.

Code:

```
def bubble_sort(arr):
    n = len(arr)
    for i in range(n):
        swapped = False
        for j in range(0, n-i-1):
            if arr[j] > arr[j+1]:
                arr[j], arr[j+1] = arr[j+1], arr[j]
                swapped = True
        # If no two elements were swapped by the inner loop, then break
        if not swapped:
            break
    return arr
```

```
random_array = [5, 2, 9, 1, 5, 6]
print("Random Array - Before Sorting:", random_array)
sorted_random_array = bubble_sort(random_array)
print("Random Array - After Sorting:", sorted_random_array)
```

```
reverse_sorted_array = [10, 8, 6, 4, 2]
print("Reverse Sorted Array - Before Sorting:", reverse_sorted_array)
sorted_reverse_array = bubble_sort(reverse_sorted_array)
print("Reverse Sorted Array - After Sorting:", sorted_reverse_array)
```

```
sorted_array = [1, 2, 3, 4, 5]
print("Already Sorted Array - Before Sorting:", sorted_array)
sorted_sorted_array = bubble_sort(sorted_array)
print("Already Sorted Array - After Sorting:", sorted_sorted_array)
```

output:

```
PS C:\Users\karth>
PS C:\Users\karth> & C:/Users/karth/AppData/Local/Programs/Python/Python312/python.exe c:/Users/karth/OneDrive/Documents/OriginLab/problems.py
Random Array - Before Sorting: [5, 2, 9, 1, 5, 6]
Random Array - After Sorting: [1, 2, 5, 5, 6, 9]
Reverse Sorted Array - Before Sorting: [10, 8, 6, 4, 2]
Reverse Sorted Array - After Sorting: [2, 4, 6, 8, 10]
Already Sorted Array - Before Sorting: [1, 2, 3, 4, 5]
Already Sorted Array - After Sorting: [1, 2, 3, 4, 5]
PS C:\Users\karth>
```

Time complexity: $f(n)=O(n*n)$