Exp-3.5

Title:

Quick Sort with First Element as Pivot and Stepwise Array Display

Aim:

To implement Quick Sort using the first element as the pivot, display the array after each partition and recursive call until the entire array is sorted.

Algorithm

1. Start.
2. Choose the first element as the pivot.
3. Partition the array into elements less than the pivot placed before it, and elements greater after it.
4. Display the array after partitioning.
5. Recursively apply Quick Sort on the left and right sub-arrays formed by the partition.
6. Display the array after each recursive sort call.
7. Stop after the entire array is sorted.

Input:

Enter number of elements: 9

Enter the array elements: 10 16 8 12 15 6 3 9 5

Output:

Partitioned with pivot 10: 3,5,6,8,9,10,12,15,16

Program output :

```
def quick_sort(arr, low, high):
    if low < high:
        pivot_index = partition(arr, low, high)
        print(f"Array after partition with pivot {arr[pivot_index]}: {','.join(map(str, 
arr))}")
        quick_sort(arr, low, pivot_index - 1)
        quick_sort(arr, pivot_index + 1, high)


def partition(arr, low, high):
    pivot = arr[low]  # First element as pivot
    left = low + 1
    right = high# Apply Quick Sort
quick_sort(a, 0, N - 1)


print("Sorted array:", ",".join(map(str, a)))


    done = False
    while not done:
        while left <= right and arr[left] <= pivot:
            left = left + 1
        while arr[right] >= pivot and right >= left:
            right = right - 1
        if right < left:
            done = True
        else:
            arr[left], arr[right] = arr[right], arr[left]
    arr[low], arr[right] = arr[right], arr[low]
    return right


N = int(input("Enter number of elements: "))
a = list(map(int, input("Enter the array elements: ").split()))


quick_sort(a, 0, N - 1)


print("Sorted array:", ",".join(map(str, a)))
```

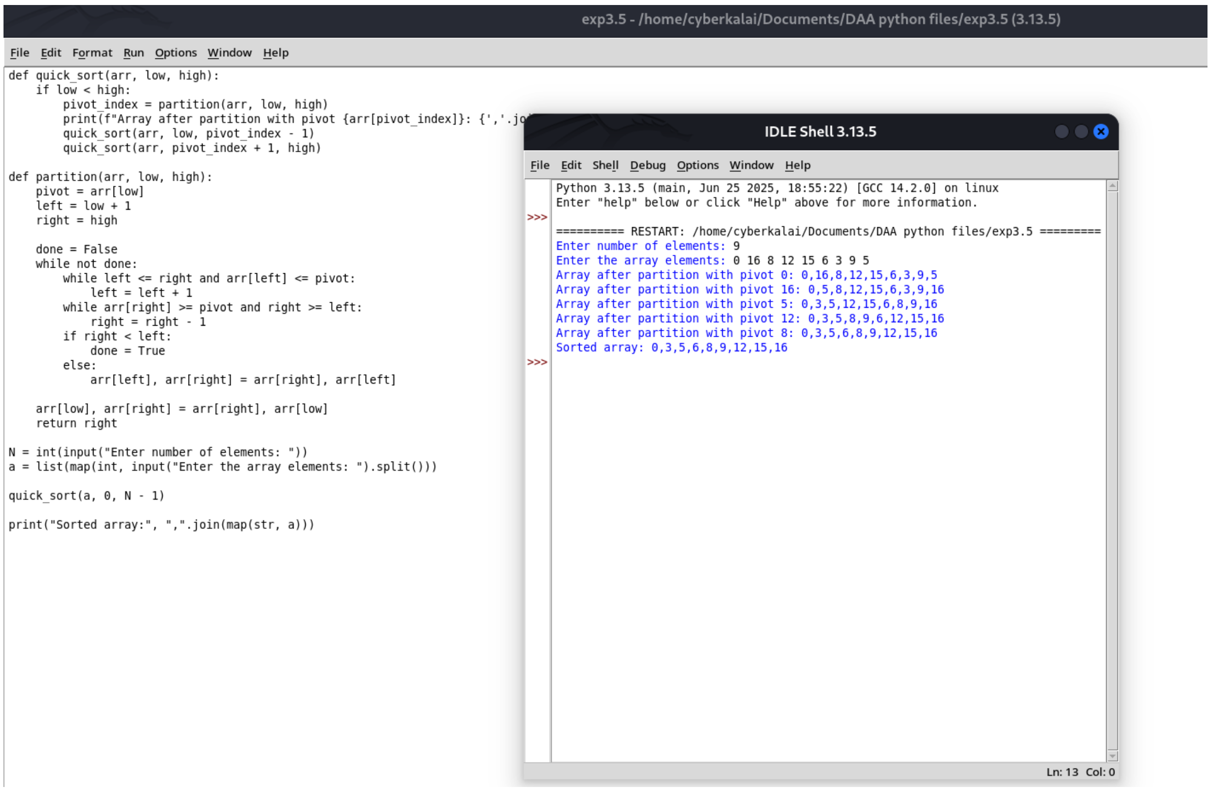
Performance Analysis:

Time Complexity: O(n log n)

Space Complexity: O(log n)

Program Output:

Thus, the Quick Sort implementation with first pivot selection and array display after each recursive call works as required.


exp3.5 - /home/cyberkalai/Documents/DAA python files/exp3.5 (3.13.5)
File Edit Format Run Options Window Help
IDLE Shell 3.13.5
File Edit Shell Debug Options Window Help
Ln: 13 Col: 0

```python
def quick_sort(arr, low, high):
    if low < high:
        pivot_index = partition(arr, low, high)
        print(f"Array after partition with pivot {arr[pivot_index]}: {','.jo
        quick_sort(arr, low, pivot_index - 1)
        quick_sort(arr, pivot_index + 1, high)

def partition(arr, low, high):
    pivot = arr[low]
    left = low + 1
    right = high

    done = False
    while not done:
        while left <= right and arr[left] <= pivot:
            left = left + 1
        while arr[right] >= pivot and right >= left:
            right = right - 1
        if right < left:
            done = True
        else:
            arr[left], arr[right] = arr[right], arr[left]

    arr[low], arr[right] = arr[right], arr[low]
    return right

N = int(input("Enter number of elements: "))
a = list(map(int, input("Enter the array elements: ").split()))

quick_sort(a, 0, N - 1)

print("Sorted array:", ",".join(map(str, a)))
```

```
Python 3.13.5 (main, Jun 25 2025, 18:55:22) [GCC 14.2.0] on linux
Enter "help" below or click "Help" above for more information.
>>>
========== RESTART: /home/cyberkalai/Documents/DAA python files/exp3.5 =========
Enter number of elements: 9
Enter the array elements: 0 16 8 12 15 6 3 9 5
Array after partition with pivot 0: 0,16,8,12,15,6,3,9,5
Array after partition with pivot 16: 0,5,8,12,15,6,3,9,16
Array after partition with pivot 5: 0,3,5,12,15,6,8,9,16
Array after partition with pivot 12: 0,3,5,8,9,6,12,15,16
Array after partition with pivot 8: 0,3,5,6,8,9,12,15,16
Sorted array: 0,3,5,6,8,9,12,15,16
>>>
```

Result:

Thus, the modified Merge Sort program executed successfully with comparison counting.