

123) Sort without using builtin function

CODE:

```
def merge_sort(arr):
    if len(arr) <= 1:
        return arr

    mid = len(arr) // 2
    left = merge_sort(arr[:mid])
    right = merge_sort(arr[mid:])

    return merge(left, right)

def merge(left, right):
    result = []
    i = j = 0

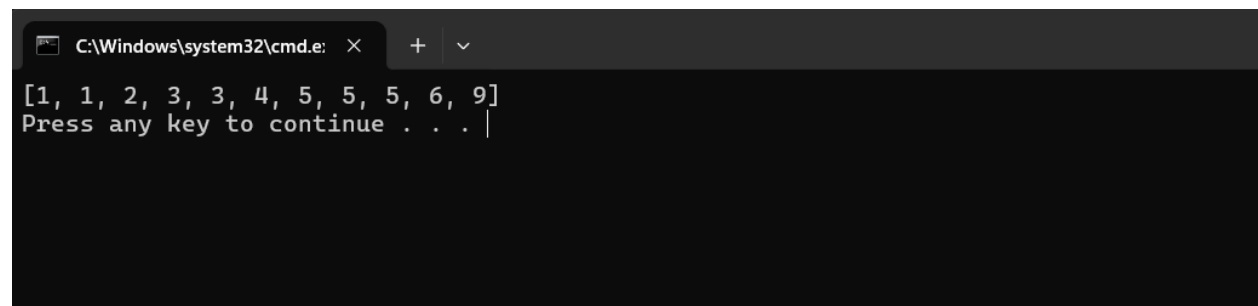
    while i < len(left) and j < len(right):
        if left[i] < right[j]:
            result.append(left[i])
            i += 1
        else:
            result.append(right[j])
            j += 1

    result.extend(left[i:])
    result.extend(right[j:])

    return result

nums = [3, 1, 4, 1, 5, 9, 2, 6, 5, 3, 5]
sorted_nums = merge_sort(nums)
print(sorted_nums)
```

OUTPUT:

A screenshot of a Windows command prompt window. The title bar shows 'C:\Windows\system32\cmd.e' with a close button. The window contains the output of the program: '[1, 1, 2, 3, 3, 4, 5, 5, 5, 6, 9]' followed by a prompt 'Press any key to continue . . . |'.

TIME COMPLEXITY :  $O(n \log n)$