Iterative merge sort

```
def merge(left, right):
    if not len(left) or not len(right):
        return left or right

    result = []
    i, j = 0, 0
    while (len(result) < len(left) + len(right)):
        if left[i] < right[j]:
            result.append(left[i])
            i+= 1
        else:
            result.append(right[j])
            j+= 1
        if i == len(left) or j == len(right):
            result.extend(left[i:] or right[j:])
            break

    return result

def mergesort(list):
    if len(list) < 2:
        return list

    middle = len(list)/2
    left = mergesort(list[:middle])
    right = mergesort(list[middle:])

    return merge(left, right)

seq = [12, 11, 13, 5, 6, 7]
print("Given array is")
print(seq);
print("\n")
print("Sorted array is")
print(mergesort(seq))
```

Iterative Quick Sort

```
def partition(arr, low, high):
    i = (low - 1)
    pivot = arr[high]

    for j in range(low, high):

        if arr[j] <= pivot:

            i += 1
            arr[i], arr[j] = arr[j], arr[i]

    arr[i + 1], arr[high] = arr[high], arr[i + 1]
    return (i + 1)

def quickSort(arr, low, high):
    if low < high:

        pi = partition(arr, low, high)
        quickSort(arr, low, pi-1)
        quickSort(arr, pi + 1, high)

if __name__ == '__main__' :

    arr = [4, 2, 6, 9, 2]
    n = len(arr)

    quickSort(arr, 0, n - 1)

    for i in range(n):
        print(arr[i], end = " ")
```