In-Place Merge Sort

Approach 1:

- Maintain two pointers which point to start of the segments which have to be merged.
- Compare the elements at which the pointers are present.
- If *element1* < *element2* then *element1* is at right position, simply increase *pointer1*.
- Else shift all the elements between element1 and *element2(including element1 but excluding element2)* right by 1 and then place the element2 in the previous place*(i.e. before shifting right)* of element1. Increment all the pointers by *1*.

Below is the implementation of the above approach:

```
def merge(arr, start, mid, end):
    start2 = mid + 1

    # If the direct merge is already sorted
    if (arr[mid] <= arr[start2]):
        return

    # Two pointers to maintain start
    # of both arrays to merge
    while (start <= mid and start2 <= end):

        # If element 1 is in right place
        if (arr[start] <= arr[start2]):
            start += 1
        else:
            value = arr[start2]
            index = start2

            # Shift all the elements between element 1
            # element 2, right by 1.
            while (index != start):
                arr[index] = arr[index - 1]
                index -= 1

            arr[start] = value

            # Update all the pointers
            start += 1
```

```
            mid += 1
            start2 += 1


'''
* l is for left index and r is right index of
the sub-array of arr to be sorted
'''


def mergeSort(arr, l, r):
    if (l < r):

        # Same as (l + r) / 2, but avoids overflow
        # for large l and r
        m = l + (r - l) // 2

        # Sort first and second halves
        mergeSort(arr, l, m)
        mergeSort(arr, m + 1, r)

        merge(arr, l, m, r)


''' UTILITY FUNCTIONS '''
''' Function to pran array '''


def printArray(A, size):

    for i in range(size):
        print(A[i], end=" ")
    print()


''' Driver program to test above functions '''
if __name__ == '__main__':
    arr = [12, 11, 13, 5, 6, 7]
    arr_size = len(arr)

    mergeSort(arr, 0, arr_size - 1)
    printArray(arr, arr_size)
```

Learn amd memorize