1. **Algorithm of bubble sort**

beginBubbleSort(arr)

for all array elements

if arr[i]>arr[i+1]

swap(arr[i], arr[i+1])

end if

end for

return arr

end of bubbleSort

1. **Algorithm of merge sort**

MergeSort(arr[], l, r)

If r > l

Step 1: Find the middle point to divide the array into two halves:

middle m = l+ (r-l)/2

Step 2: Call mergeSort for first half:

Call mergeSort(arr, l, m)

Step **3.** Call mergeSort for second half:

Call mergeSort(arr, m+1, r)

Step **4.** Merge the two halves sorted in step 2 and 3:

Call merge(arr, l, m, r)

1. **Algorithm of Quick sort**
2. QUICKSORT (array A, start, end)
3. {
4. 1 **if** (start < end)
5. 2 {
6. 3 p = partition(A, start, end)
7. 4 QUICKSORT (A, start, p - 1)
8. 5 QUICKSORT (A, p + 1, end)
9. 6 }
10. }

For partition algorithm

1. PARTITION (array A, start, end)
2. {
3. 1 pivot ? A[end]
4. 2 i ? start-1
5. 3 **for** j ? start to end -1 {
6. 4 **do** **if** (A[j] < pivot) {
7. 5 then i ? i + 1
8. 6 swap A[i] with A[j]
9. 7  }}
10. 8 swap A[i+1] with A[end]
11. 9 **return** i+1
12. }