Mergesort

Merge Sort

- Merge sort is a comparison sorting technique.
- This technique follows the divide-and-conquer approach.
- It maintains the following 3 steps:

  1. Divide: Divide N-element sequence to be sorted into two subsequences of about N/2 elements each and sort the two subsequences recursively .

  2. Conquer: Merge the two sorted subsequences to produce the sorted result.

- Merge sort uses the “merge” step to combine two sorted sublists to create one single sorted list.
- Suppose A is a sorted list with R elements and B is another sorted list with S elements. After merging there is only a single sorted list C with N=R+S elements.

Mergesort Algorithm

Mergesort(List, N)

(1) Msort (List, TempList, 0, N-1)

(2) End

Msort(List, TempList, Left, Right)

(1) If Left < Right do steps 2 to 5

(2) Set Center = (Left+Right)/2

(3) Msort (List, Temp List,Left,Center)

(4) Msort (List,TempList,Center+1,Right)

(5) Merge(List,TempList,Left,Center+1,Right)

(6) End

Merge(List,TempList,Lpos,Rpos,RightEnd)

(1) Set LeftEnd = Rpos-1 and TmpPos = Lpos

(2) NumElement = RightEnd – Lpos + 1

(3) While Lpos<=LeftEnd && Rpos<=RightEnd

(4) If List [Lpos] <= List [Rpos] then

(5) TmpList[TmpPos++]=List [Lpos++]

(6) Else

(7) TmpList[TmpPos++]=List [Rpos++]

(8) While Lpos <= LeftEnd

(7) TmpList[TmpPos++] = List [Lpos++]

(10) While Rpos <= RightEnd

(11) TmpList[TmpPos++] = List [Rpos++]

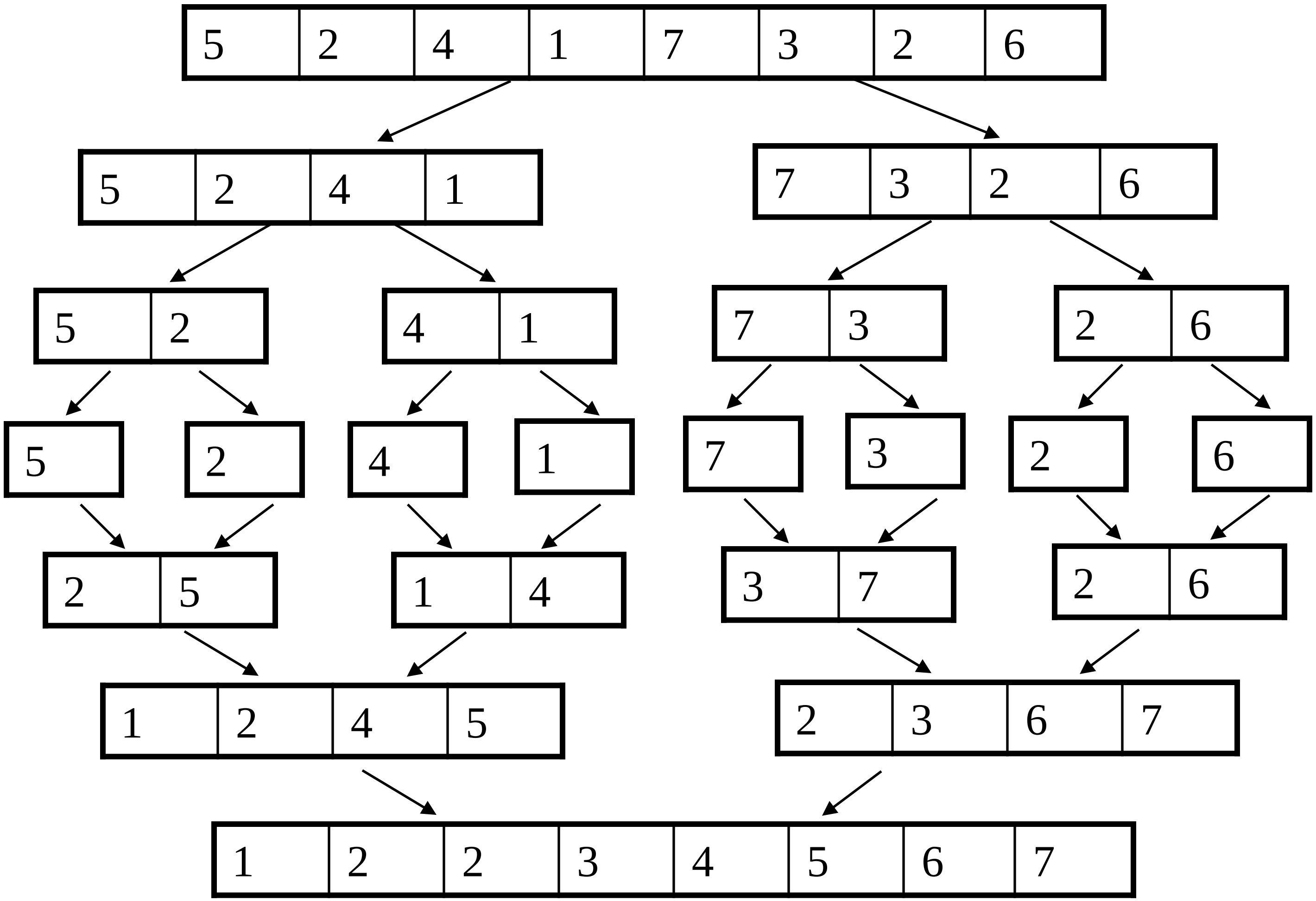
(12) For I = 0 to NumElement do step 13

(13) List [RightEnd--]=TmpList [RightEnd--]

(14) End

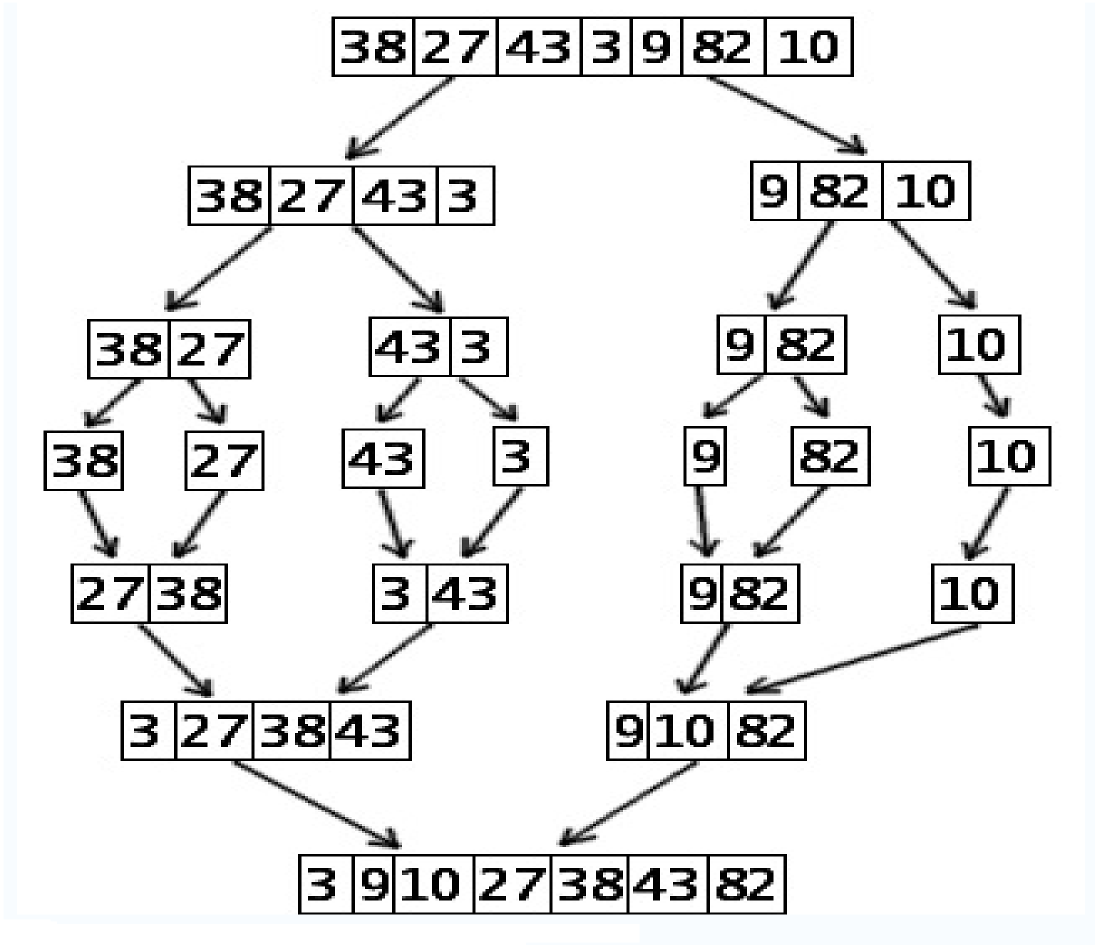
Example:

Suppose Array A = 5, 2, 4, 7, 1, 3, 2, 6. Sort the array using Mergesort.



Example:

Suppose Array A = 38, 27, 43, 3, 9, 82 and 10. Sort the array using Mergesort.



Complexity of Merge-Sort

Let T(N) be the number of comparisons needed to sort N elements using merge sort. This algorithm requires at most logN passes. Moreover, each pass merges a total of N elements and each pass will require at most N comparisons. So, for all cases, T(N) = 0(NlogN).

Recurrence Relation for Mergesort

$T(1) = 1$ For $N = 1$

$T(N) = 2T(N/2) + N$ Otherwise

$T(N) = 2T(N/2) + N$ is equivalent to $O(N\log_2 N)$

Solution:

$T(N) \quad = \; 2T(N/2) + N$

$T(N/2) = \; 2T(N/4) + N/2$

$T(N/4) = \; 2T(N/8) + N/4$

……………………..

$T(2) \quad = \; 2T(1) + 2$

$T(N) = 2^K T(N/2^K) + K.N$

By using $2^K = N$, it is obtained as

$T(N) = NT(1) + NLog_2N = N + NLog_2N = O(NLog_2N)$

So, $T(N) = 2T(N/2) + N$ is equivalent to $O(Nlog_2N)$.

END