

## Episode 28: Merge-Sort (Mechanics)

rev. 25/02/2008 13:29

---

**Reading:** GT 4.1.0–4.1.1, pp. 218–222 [up to 40 minutes]

**Ingredients:** MERGE, MERGE-SORT

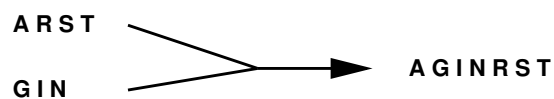
---

Divide-and-conquer is a popular design principle.  
MERGE-SORT is an efficient sorting algorithm  
that divides and conquers.

## MERGE

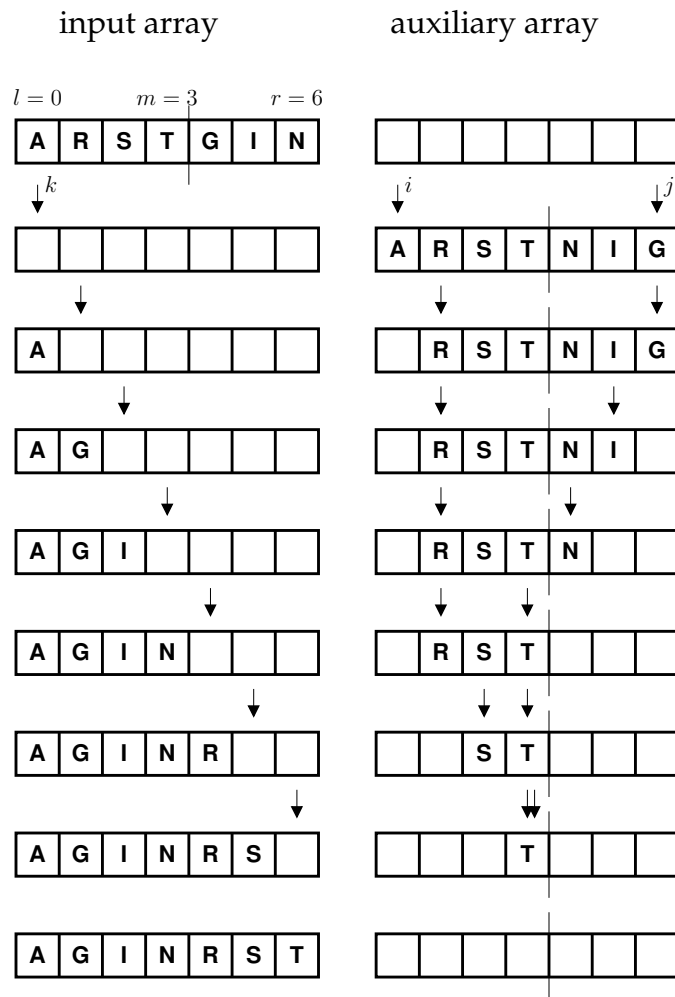
Merge is an operation that given two sorted sequences (files/arrays/lists) weaves them into a single sorted sequence (file/array/list).

Example:

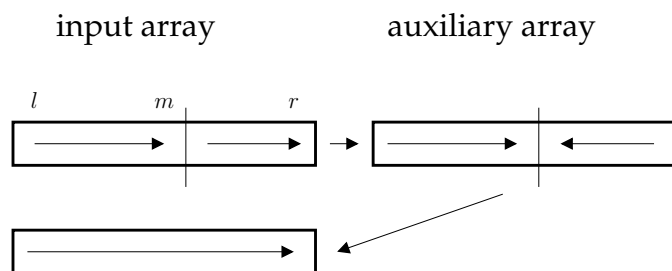


There are many ways to implement merge. For arrays we often copy two input arrays into an auxiliary array. The second one we copy backwards so that its largest element acts as a *sentinel* for the first array.

Example:  $\text{MERGE}(a, 0, 3, 6)$  where  $a = \{A R S T G I N\}$ .



Or in general



MERGE( $a, l, m, r$ )

▷ Assume existence of array  $aux$ , the same size as  $a$

```

1  for  $i \leftarrow l$  to  $m$  do  $aux[i] \leftarrow a[i]$ 
2  for  $j \leftarrow m + 1$  to  $r$  do  $aux[r - (j - m - 1)] \leftarrow a[j]$ 
3   $i \leftarrow l, j \leftarrow r$ 
4  for  $k \leftarrow l$  to  $r$ 
5      do if  $aux[j] < aux[i]$ 
6          then  $a[k] \leftarrow aux[j--]$ 
7          else  $a[k] \leftarrow aux[i++]$ 

```

The running time of the above procedure is  $\theta(N)$ , where  $N = r - l + 1$ .

Observe that MERGE can be easily implemented also for sorted linked lists or sorted files. For files a new file is produced. For lists the algorithm merges two lists producing a third one (typically input lists are destroyed).

Applications of MERGE:

- Adding data to a sorted array: sort the data added and merge into a bigger array in linear time.
- Sorting: MERGE-SORT and external sorts.

## MERGE-SORT

MERGE-SORT uses the venerable *divide-and-conquer* design principle. It splits the sorted array in half and recursively calls itself on both parts, until parts become single element. Then it merges sorted parts using MERGE.

MERGE-SORT( $a, l, r$ )

```

1  if  $r \leq l$  return
2   $m \leftarrow \lfloor \frac{r+l}{2} \rfloor$ 
3  MERGE-SORT( $a, l, m$ )
4  MERGE-SORT( $a, m + 1, r$ )
5  MERGE( $a, l, m, r$ )

```

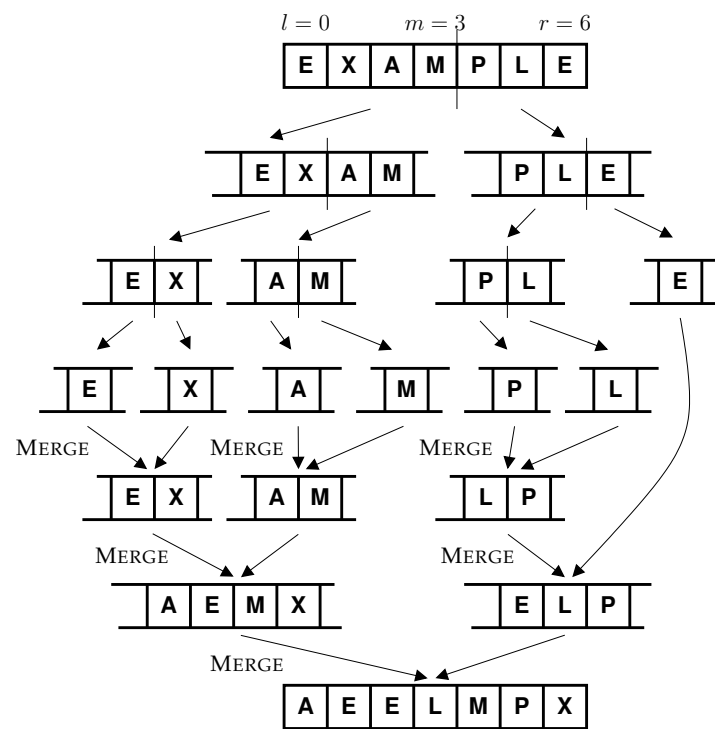


Figure 1: Running MERGE-SORT on array containing chars E X A M P L E.