

L8

Merge Sort

Algorithm

Merge Sort

- Follows D & C paradigm
- Divide: Divide the n -element array into two subarrays of size $n/2$
- Conquer: Sort the two subarrays recursively
- Combine: Merge the two sorted subarrays to produce the sorted array

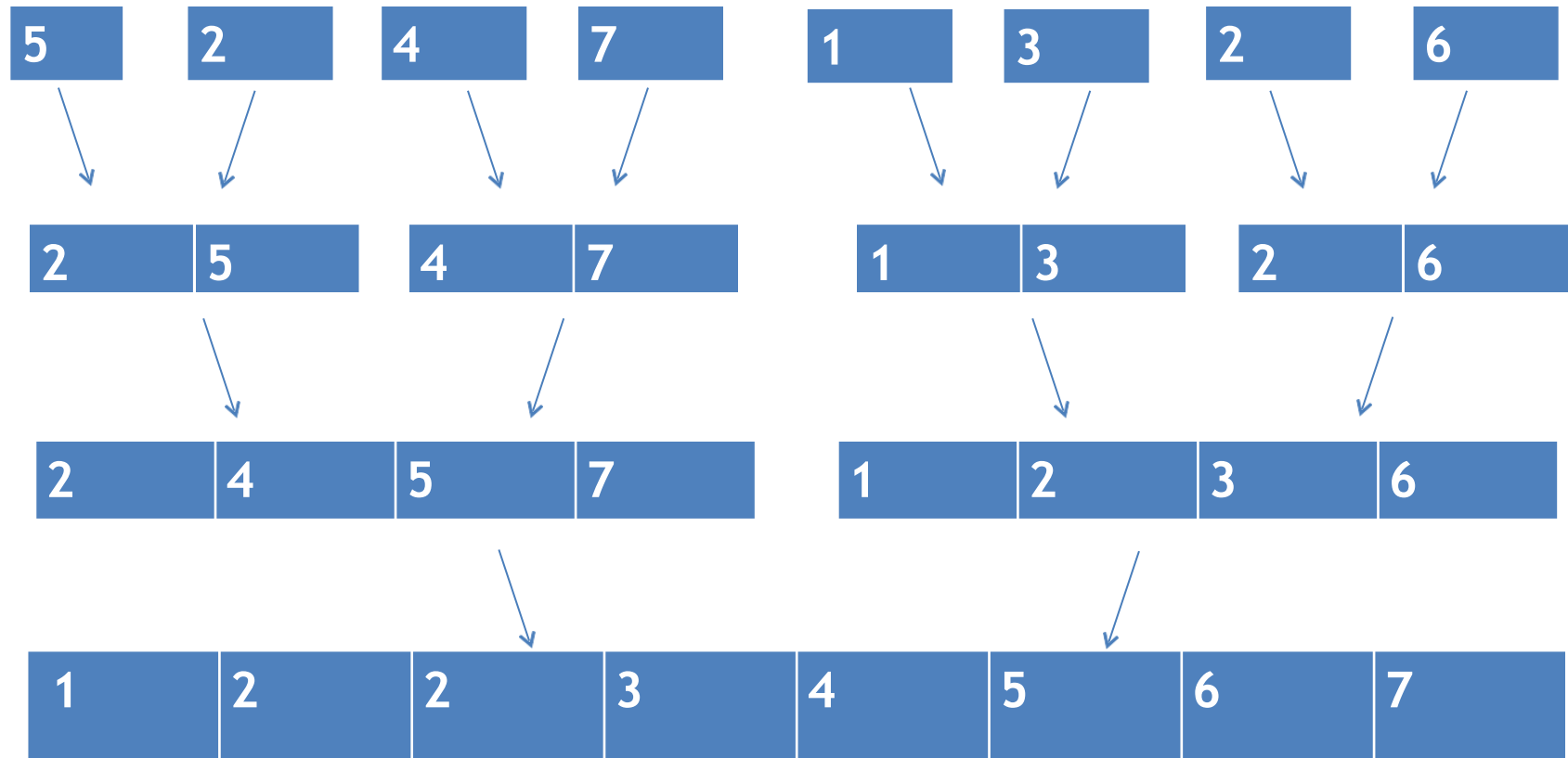
Merge Sort - Example

The operation of merge sort on the array

$A = \{5, 2, 4, 7, 1, 3, 2, 6\}$



Merging of sorted subarrays



Merge Sort - Recursive Algorithm

MERGE-SORT(A, p, r)

1 **if** $p < r$

2 $q = \lfloor (p + r) / 2 \rfloor$

3 MERGE-SORT(A, p, q)

4 MERGE-SORT($A, q + 1, r$)

5 MERGE(A, p, q, r)

MERGE-SORT

- If $p \geq r$, the subarray has at most one element and is therefore already sorted.
- Otherwise, the divide step, computes an index q that partitions $A[p \dots r]$ into two subarrays $A[p \dots q]$ and $A[q+1 \dots r]$ containing $(n/2)$ elements

MERGE-SORT(A, p, r)

```
1  if  $p < r$ 
2       $q = \lfloor (p + r) / 2 \rfloor$ 
3      MERGE-SORT( $A, p, q$ )
4      MERGE-SORT( $A, q + 1, r$ )
5      MERGE( $A, p, q, r$ )
```

Function call:

MERGE-SORT($A, 1, A.length$)

Merge sort – Recursive algorithm

- **Base case:**
 - When the size of the subproblem is 1, we don't need to do any further calls
 - Its already sorted
- **Key operation:** Merging of two sorted arrays in the combine step
- Merge is done by calling another function
Merge (A,p,q,r)

Merge function

- Merge is done by calling another function
Merge (A,p,q,r)

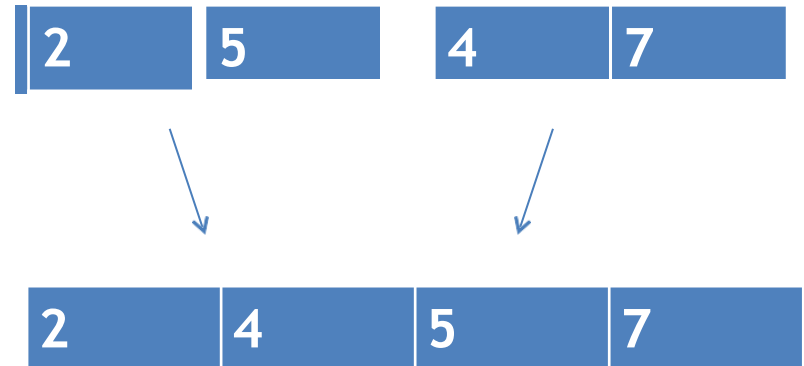
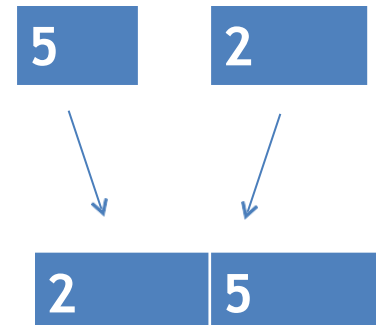
A- Array, p,q,r are indices s.t $p \leq q < r$

- **Assumption:** $A[p..q]$ and $A[q+1.. r]$ are in sorted order
- **Input:** Array A, indices p, q, and r
- **Output:** Merges $A[p..q]$ and $A[q+1 .. r]$ and produce a single sorted subarray $A[p..r]$

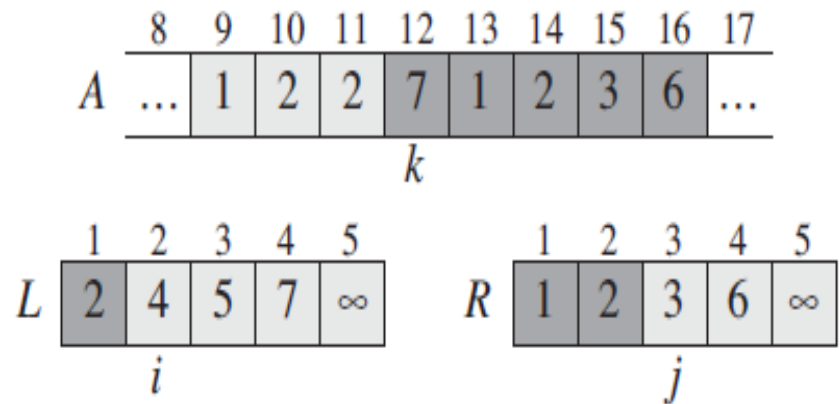
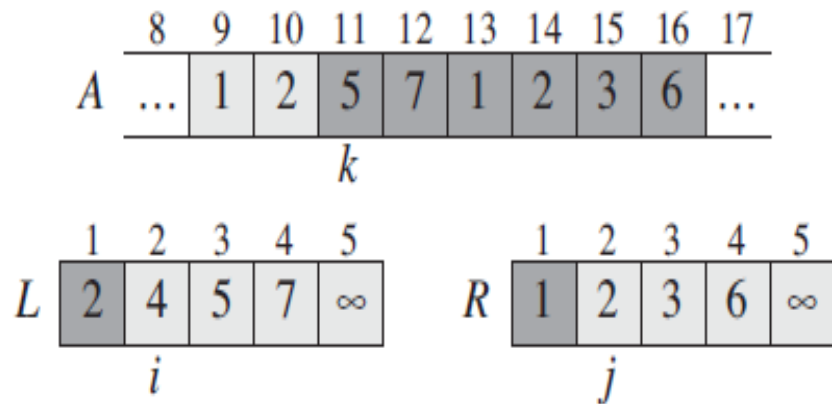
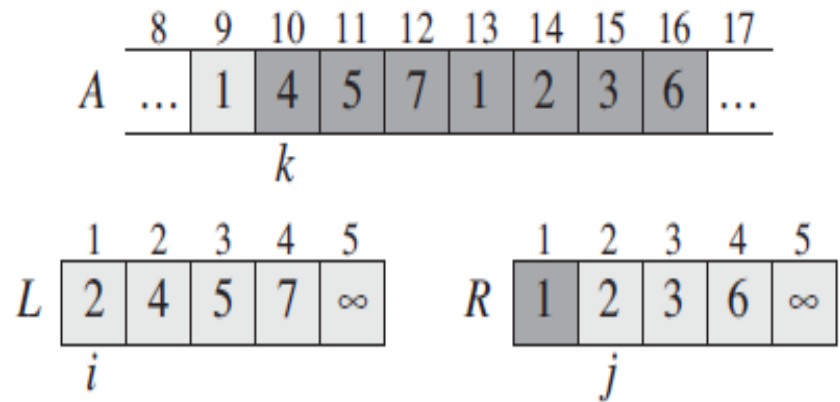
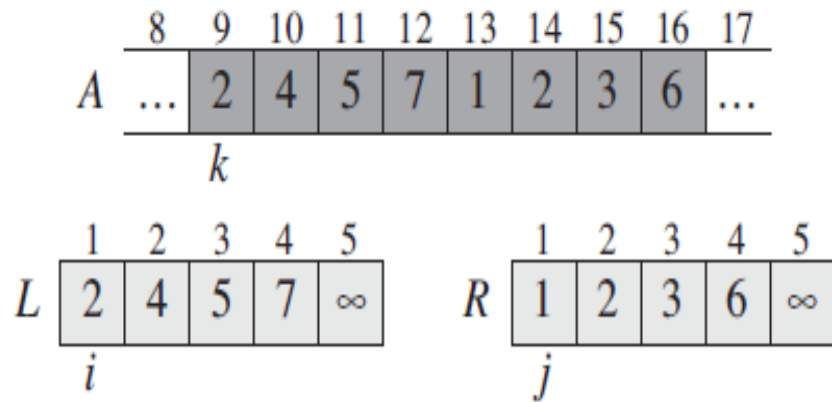
Merge function

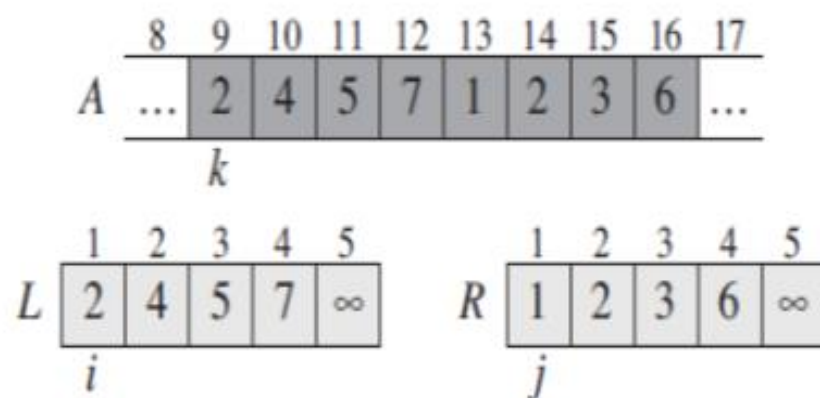
MERGE(A, p, q, r)

```
1   $n_1 = q - p + 1$ 
2   $n_2 = r - q$ 
3  let  $L[1..n_1 + 1]$  and  $R[1..n_2 + 1]$  be new arrays
4  for  $i = 1$  to  $n_1$ 
5       $L[i] = A[p + i - 1]$ 
6  for  $j = 1$  to  $n_2$ 
7       $R[j] = A[q + j]$ 
8   $L[n_1 + 1] = \infty$       Sentinel - a special value
9   $R[n_2 + 1] = \infty$ 
10  $i = 1$ 
11  $j = 1$ 
12 for  $k = p$  to  $r$ 
13     if  $L[i] \leq R[j]$ 
14          $A[k] = L[i]$ 
15          $i = i + 1$ 
16     else  $A[k] = R[j]$ 
17          $j = j + 1$ 
```

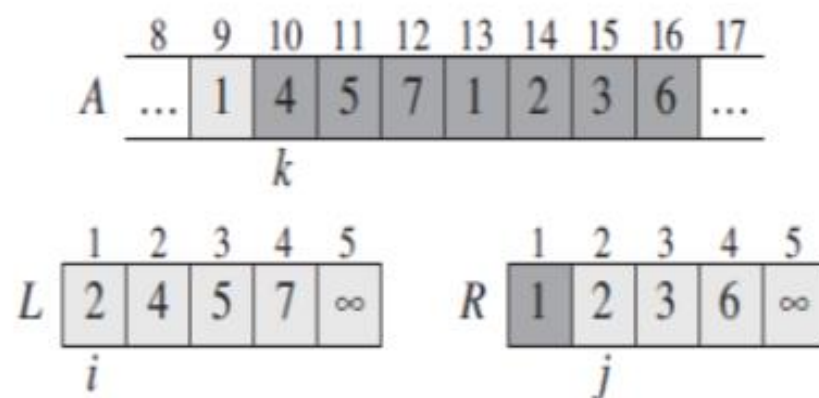


Working of Merge function





(a)



(b)

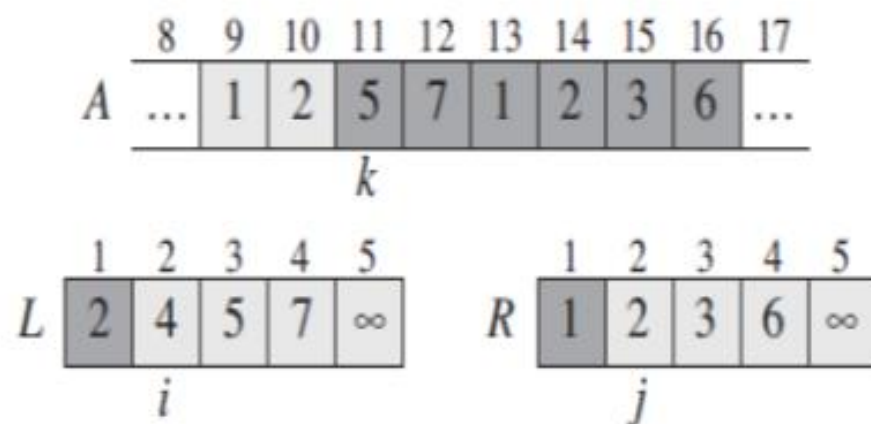
MERGE(A, p, q, r)

- 1 $n_1 = q - p + 1$
- 2 $n_2 = r - q$
- 3 let $L[1..n_1 + 1]$ and $R[1..n_2 + 1]$ be new arrays
- 4 **for** $i = 1$ **to** n_1
- 5 $L[i] = A[p + i - 1]$
- 6 **for** $j = 1$ **to** n_2
- 7 $R[j] = A[q + j]$
- 8 $L[n_1 + 1] = \infty$
- 9 $R[n_2 + 1] = \infty$

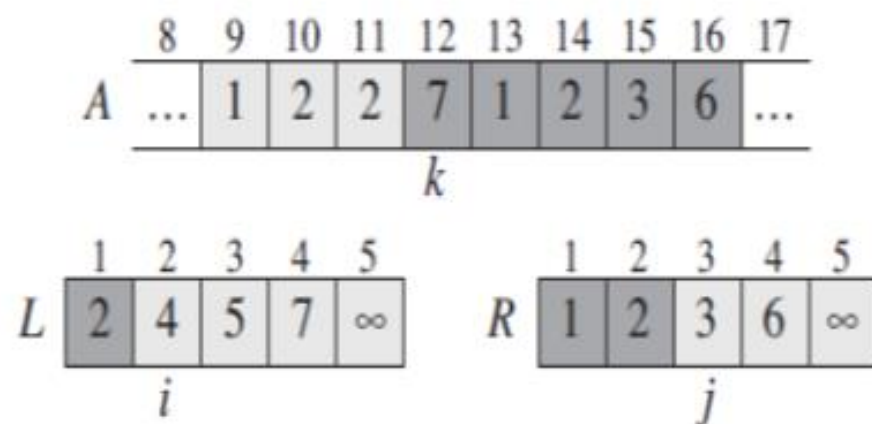
```

10   $i = 1$ 
11   $j = 1$ 
12  for  $k = p$  to  $r$ 
13      if  $L[i] \leq R[j]$ 
14           $A[k] = L[i]$ 
15           $i = i + 1$ 
16      else  $A[k] = R[j]$ 
17           $j = j + 1$ 

```



(c)



(d)

Thank You