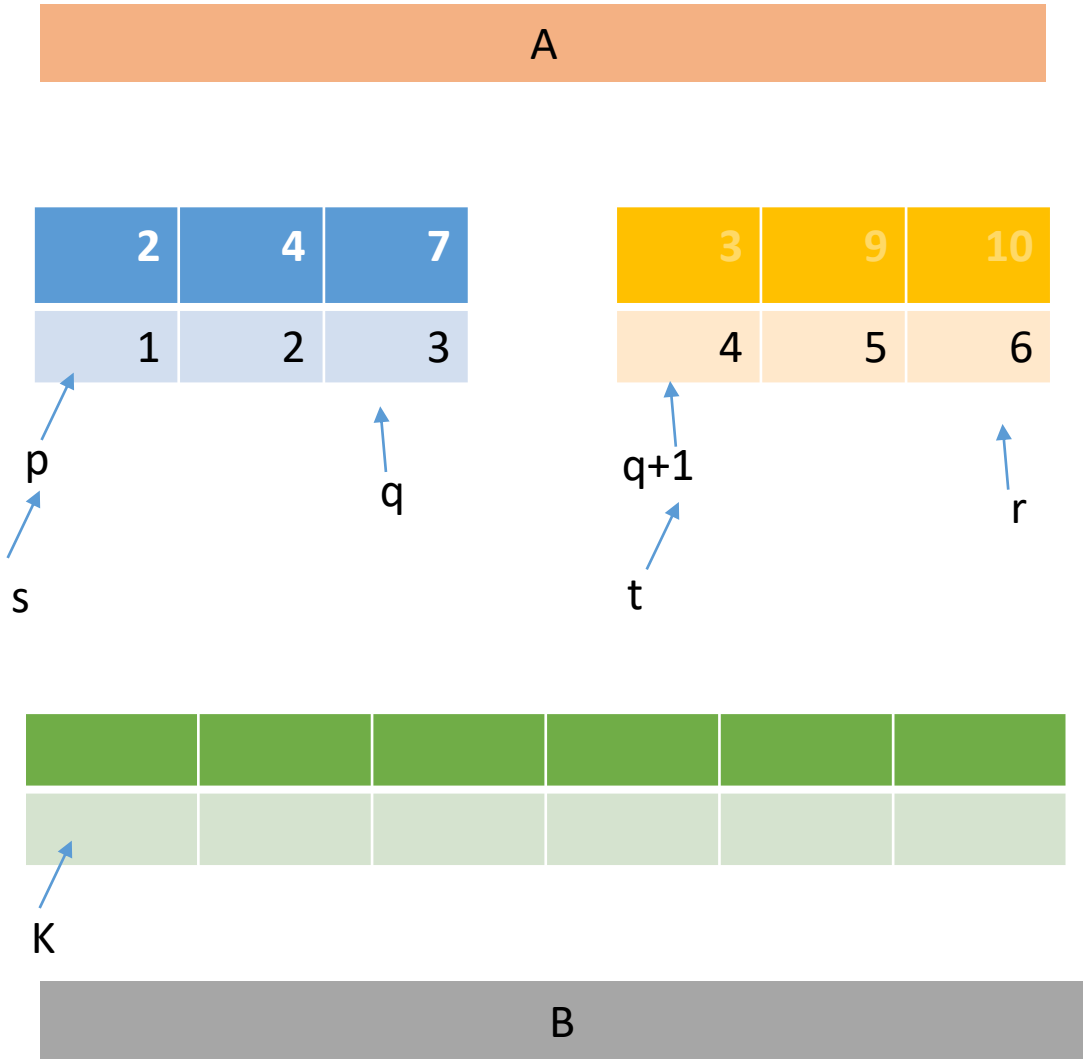
Marriage two sorted list algorithm

Algorithm: *MERGE*

Input: *An array A[0..m-1] of elements and three indices p, q and r, with* $1 \le p \le q < r \le m$, *such that both the subarrays A[p..q] and A[q + 1..r] are sorted individually in nondecreasing order.*

Output: *A[p..r] contains the result of merging the two* ***subarrays A[p..q] and A[q + 1..r].***

```
1. comment: B[p..r] is an auxiliary array.
2. s ← p; t ← q + 1; k ← p
3. while s ≤ q and t ≤ r
4.       if A[s] ≤ A[t] then
5.           B[k] ← A[s]
6.           s ← s + 1
7.       else
8.           B[k] ←A[t]
9.           t ← t + 1
10.      end if
11.      k ← k + 1
12. end while
13. if s = q + 1then B[k..r] ← A[t..r]
14. else B[k..r] ← A[s..q]
15. end if
16. A[p..r] ← B[p..r]
```



BUTTM-UP MARREGE SORTING

- **Operation of the *Bottom-up* merge sort**
  - **Operation** of the ***bottom-up* merge sort** algorithm:

    - The ***bottom-up* merge sort** algorithm ***first* merges pairs of adjacent arrays of *1* elements**
    - ***Then* merges pairs of adjacent arrays of *2* elements**
    - And ***next* merges pairs of adjacent arrays of *4* elements**
    - **And so on**....

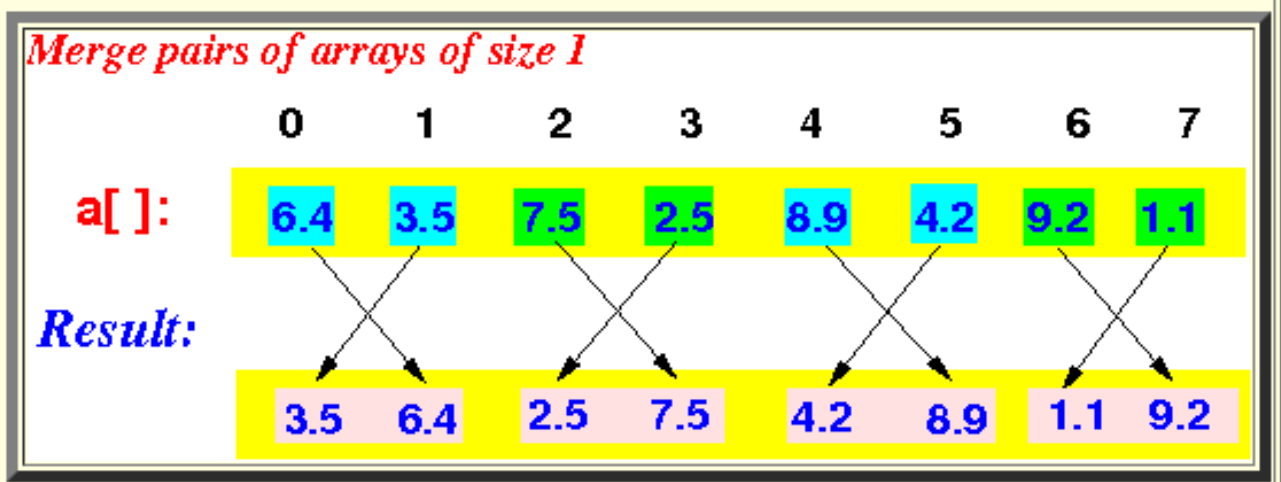
      - Until the **whole array** is **merged**

- **Input array:**

```
6.4  3.5  7.5  2.5  8.9  4.2  9.2  1.1
```

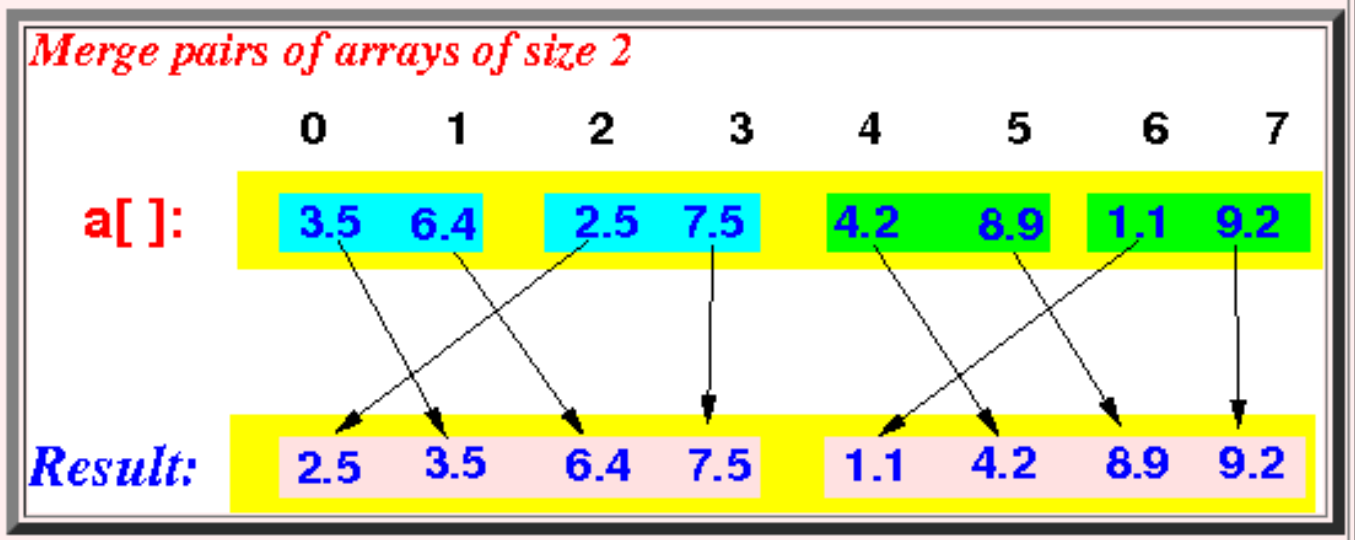
- **Iteration 1:**

  - **Merge pairs of adjacent arrays of size = 1:**



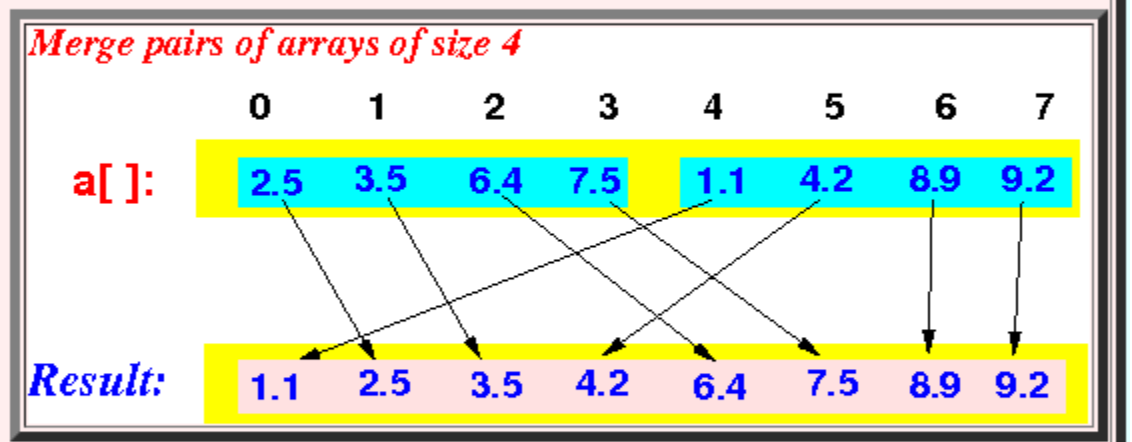
- **Iteration 2:**

  - Merge pairs of adjacent arrays of size = 2:



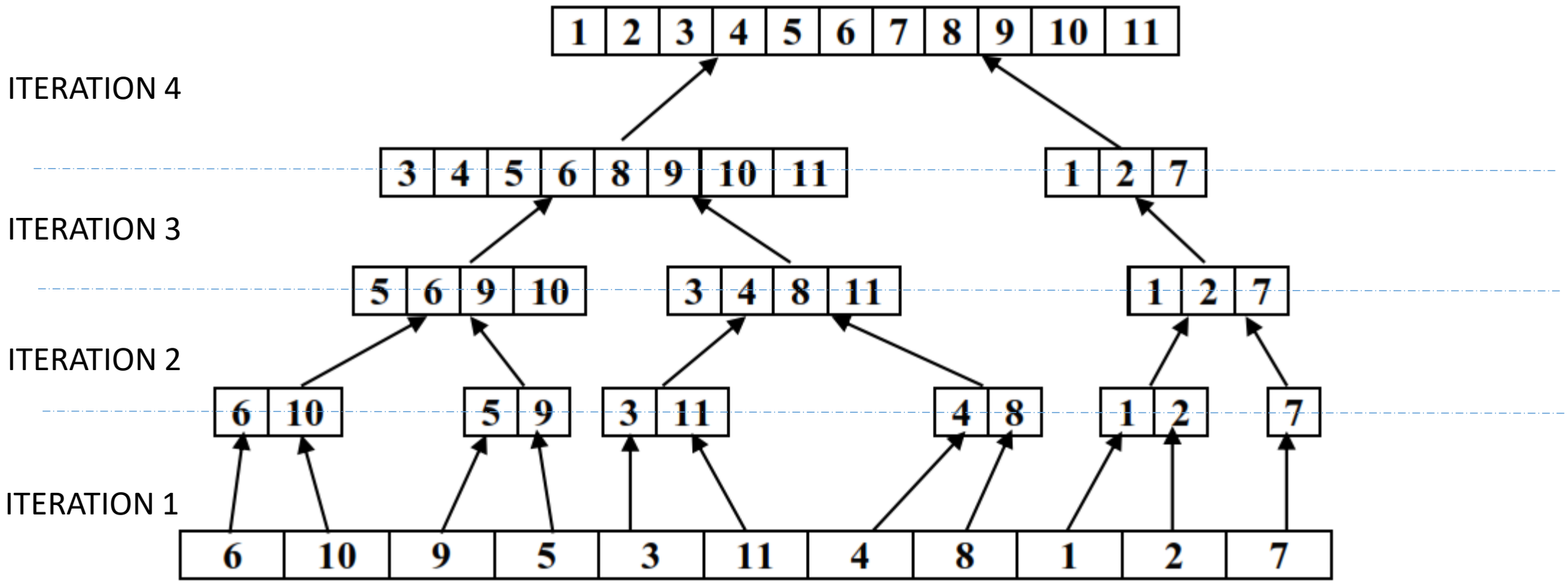
- **Iteration 3:**

  - Merge pairs of adjacent arrays of size = 4:



- The whole array has been merged

  Done !!!


ITERATION 4
ITERATION 3
ITERATION 2
ITERATION 1
1 2 3 4 5 6 7 8 9 10 11
3 4 5 6 8 9 10 11
1 2 7
5 6 9 10
3 4 8 11
1 2 7
6 10
5 9
3 11
4 8
1 2
7
6 10 9 5 3 11 4 8 1 2 7

Algorithm: *BOTTOMUPSORT*
Input: *An array A[1..n] of n elements.*
Output: *A[0..n-1] sorted in nondecreasing order.*

```
1. t ← 1
2. while     t < n-1
3.         s ← t; t ← 2s; i ← 0
4.         while i + t ≤ n-1
5.             merge(A, i  , i + s-1 , i + t -1)
6.          i ← i + t
7.         end while
8.         if i + s < n then merge(A, i , i+ s-1, n-1)
9. end while
```