

# Merging

## Merge.

- Keep track of smallest element in each sorted half.
- Insert smallest of two elements into auxiliary array.
- Repeat until done.

smallest



A	G	L	O	R
---	---	---	---	---

smallest



H	I	M	S	T
---	---	---	---	---

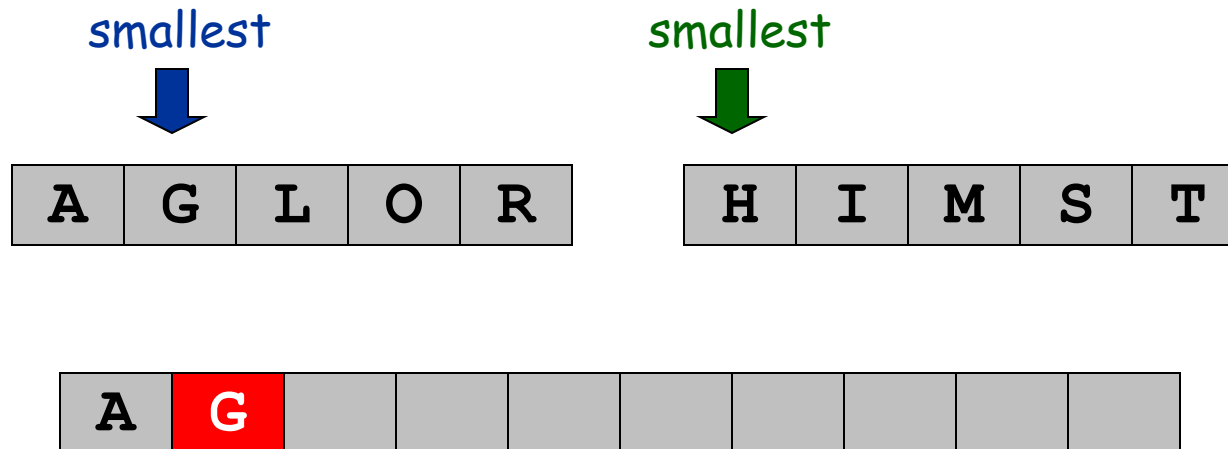
A									
---	--	--	--	--	--	--	--	--	--

auxiliary array

# Merging

## Merge.

- Keep track of smallest element in each sorted half.
- Insert smallest of two elements into auxiliary array.
- Repeat until done.

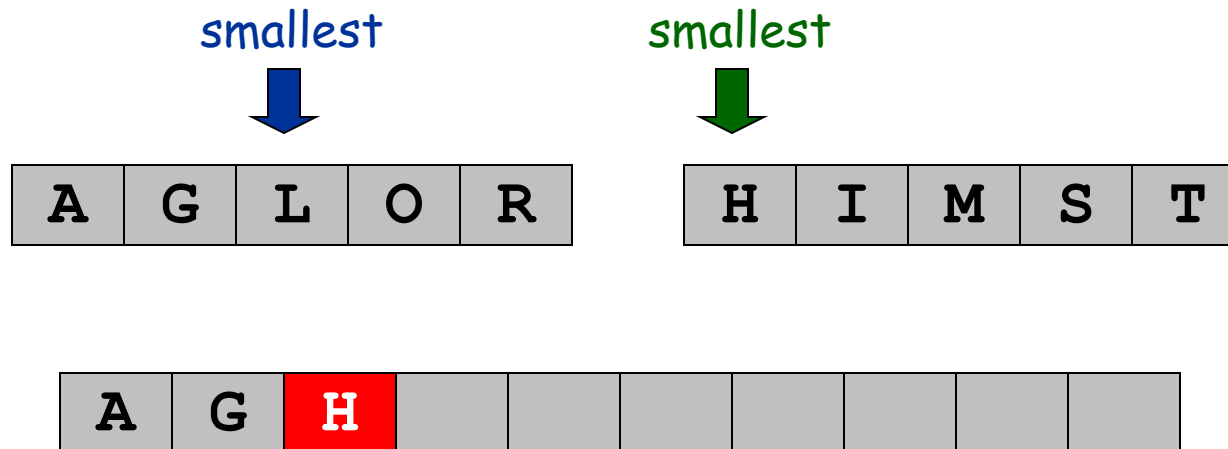


auxiliary array

# Merging

## Merge.

- Keep track of smallest element in each sorted half.
- Insert smallest of two elements into auxiliary array.
- Repeat until done.

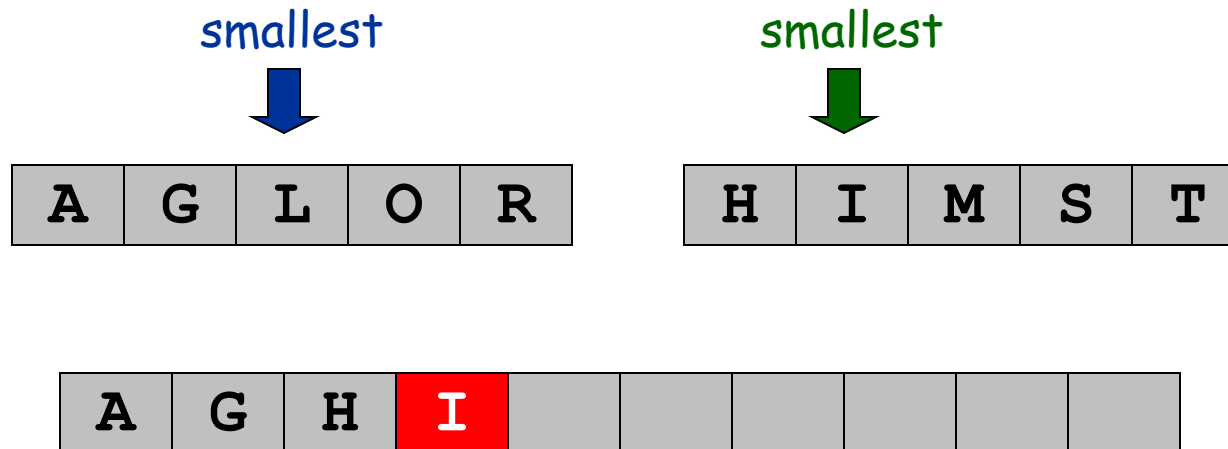


auxiliary array

# Merging

## Merge.

- Keep track of smallest element in each sorted half.
- Insert smallest of two elements into auxiliary array.
- Repeat until done.

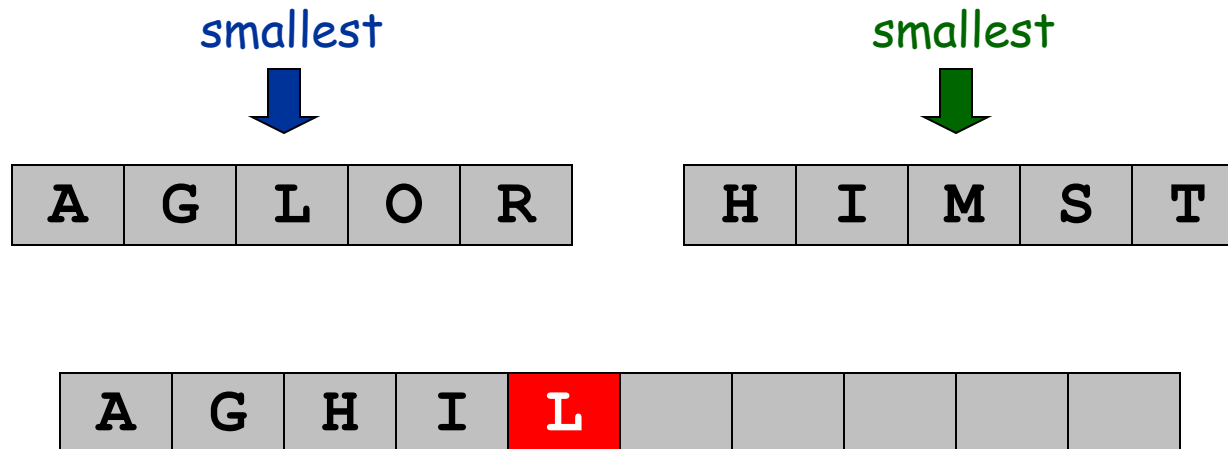


auxiliary array

# Merging

## Merge.

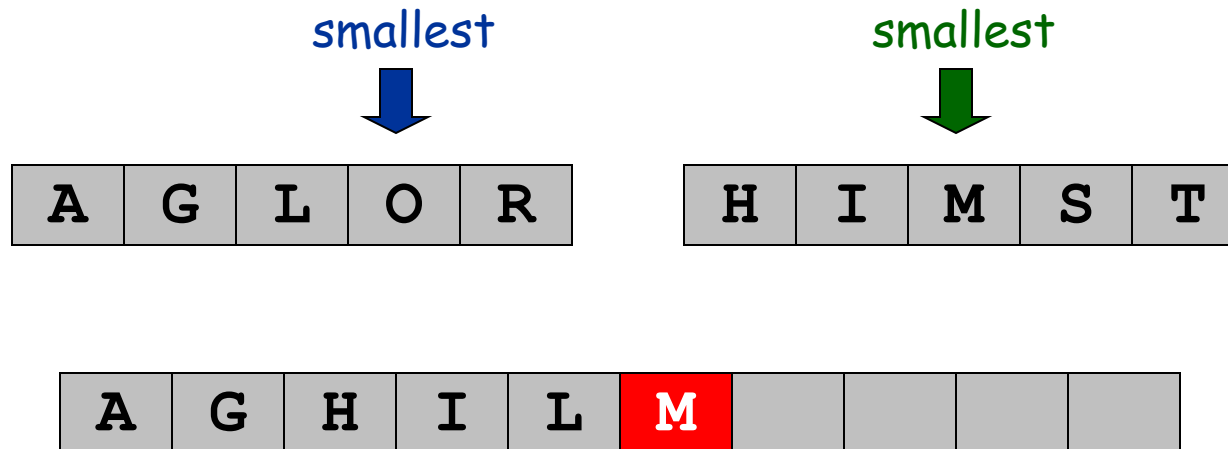
- Keep track of smallest element in each sorted half.
- Insert smallest of two elements into auxiliary array.
- Repeat until done.



# Merging

## Merge.

- Keep track of smallest element in each sorted half.
- Insert smallest of two elements into auxiliary array.
- Repeat until done.

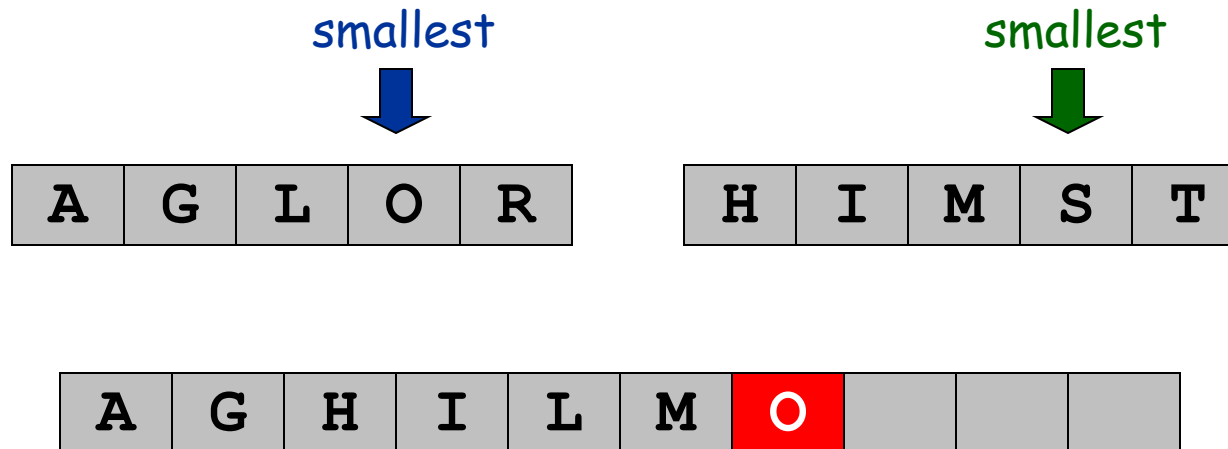


auxiliary array

# Merging

## Merge.

- Keep track of smallest element in each sorted half.
- Insert smallest of two elements into auxiliary array.
- Repeat until done.

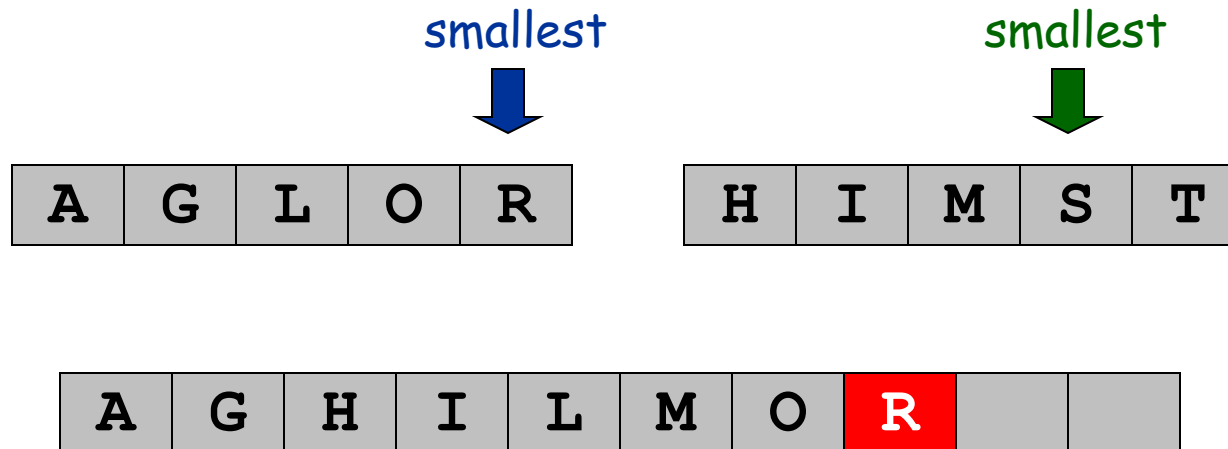


auxiliary array

# Merging

## Merge.

- Keep track of smallest element in each sorted half.
- Insert smallest of two elements into auxiliary array.
- Repeat until done.



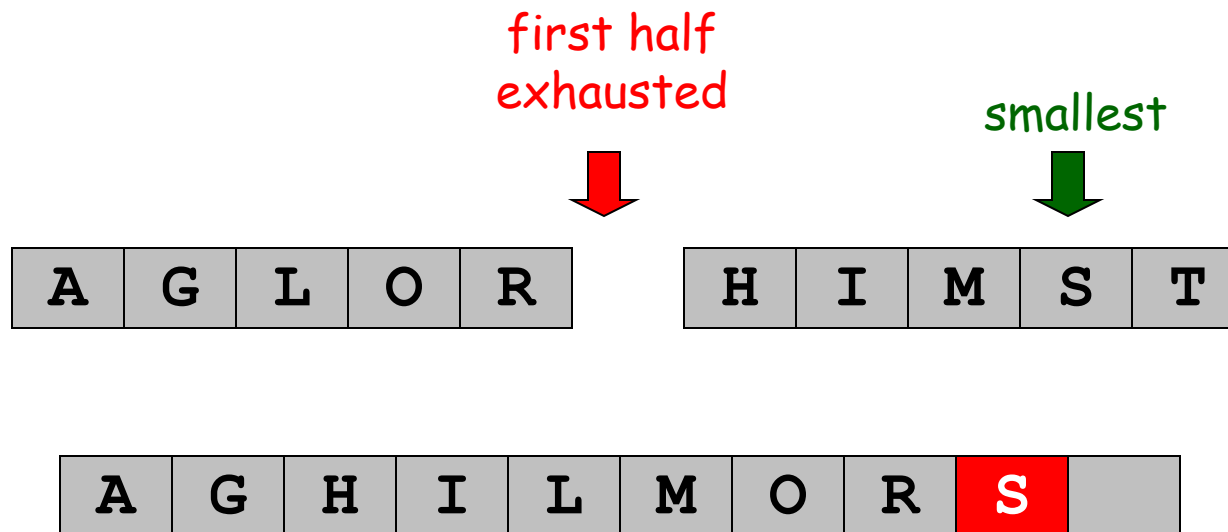
auxiliary array



# Merging

## Merge.

- Keep track of smallest element in each sorted half.
- Insert smallest of two elements into auxiliary array.
- Repeat until done.

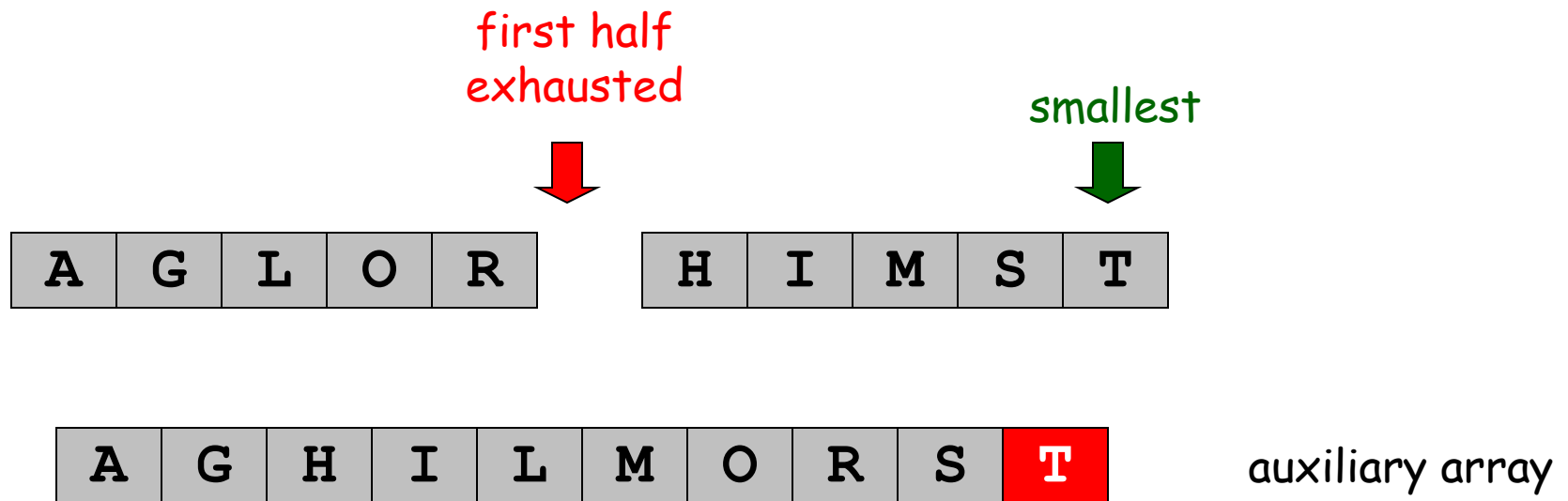


auxiliary array

# Merging

## Merge.

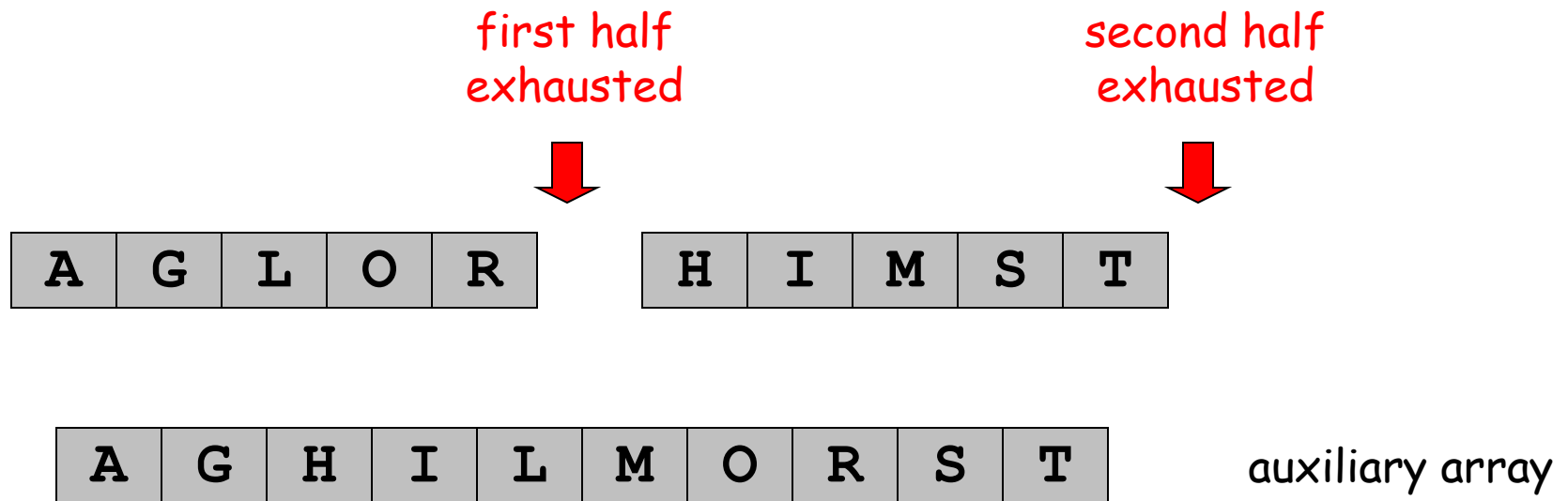
- Keep track of smallest element in each sorted half.
- Insert smallest of two elements into auxiliary array.
- Repeat until done.



# Merging

## Merge.

- Keep track of smallest element in each sorted half.
- Insert smallest of two elements into auxiliary array.
- Repeat until done.



# Merge and Count

## Merge and count step.

- Given two sorted halves, count number of inversions where  $a_i$  and  $a_j$  are in different halves.
- Combine two sorted halves into sorted whole.

$i = 6$



3	7	10	14	18	19
---	---	----	----	----	----



2	11	16	17	23	25
---	----	----	----	----	----

two sorted halves

--	--	--	--	--	--	--	--	--	--	--	--

auxiliary array

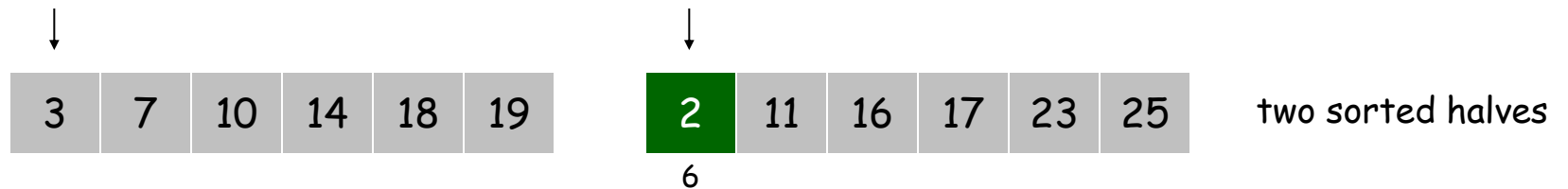
Total:

# Merge and Count

## Merge and count step.

- Given two sorted halves, count number of inversions where  $a_i$  and  $a_j$  are in different halves.
- Combine two sorted halves into sorted whole.

$i = 6$



Total: 6

# Merge and Count

## Merge and count step.

- Given two sorted halves, count number of inversions where  $a_i$  and  $a_j$  are in different halves.
- Combine two sorted halves into sorted whole.

$i = 6$



3	7	10	14	18	19
---	---	----	----	----	----



2	11	16	17	23	25
---	----	----	----	----	----

two sorted halves

6

2											
---	--	--	--	--	--	--	--	--	--	--	--

auxiliary array

Total: 6

# Merge and Count

## Merge and count step.

- Given two sorted halves, count number of inversions where  $a_i$  and  $a_j$  are in different halves.
- Combine two sorted halves into sorted whole.

$i = 6$



two sorted halves

6



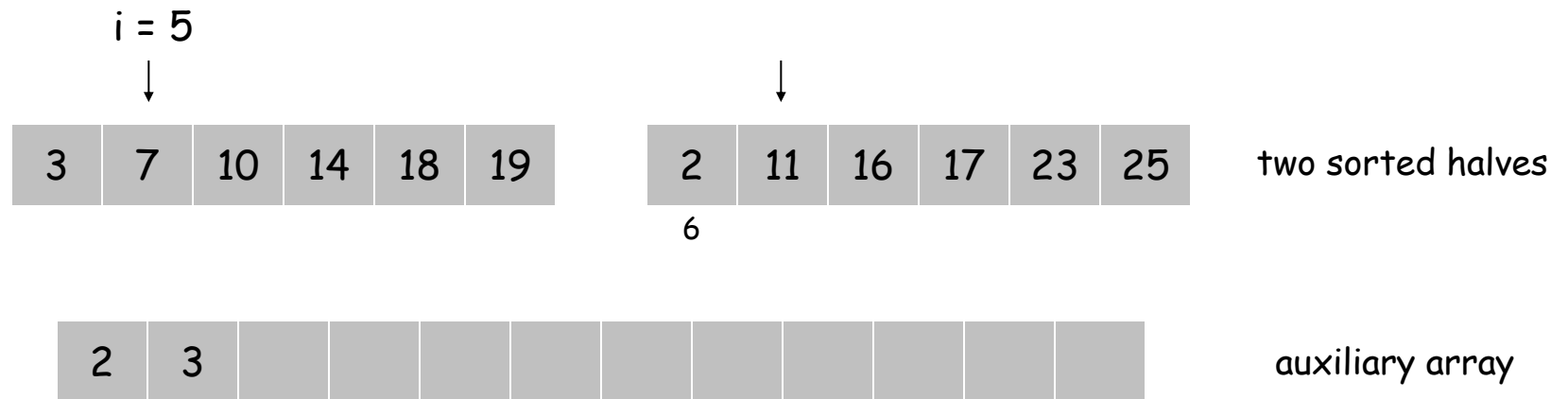
auxiliary array

Total: 6

# Merge and Count

## Merge and count step.

- Given two sorted halves, count number of inversions where  $a_i$  and  $a_j$  are in different halves.
- Combine two sorted halves into sorted whole.



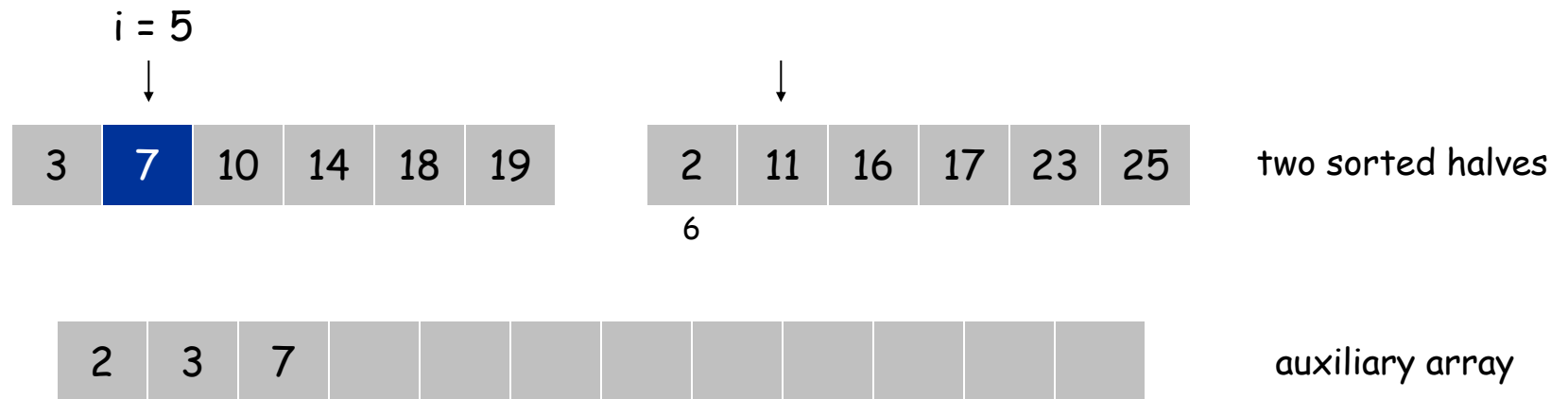
Total: 6



# Merge and Count

## Merge and count step.

- Given two sorted halves, count number of inversions where  $a_i$  and  $a_j$  are in different halves.
- Combine two sorted halves into sorted whole.

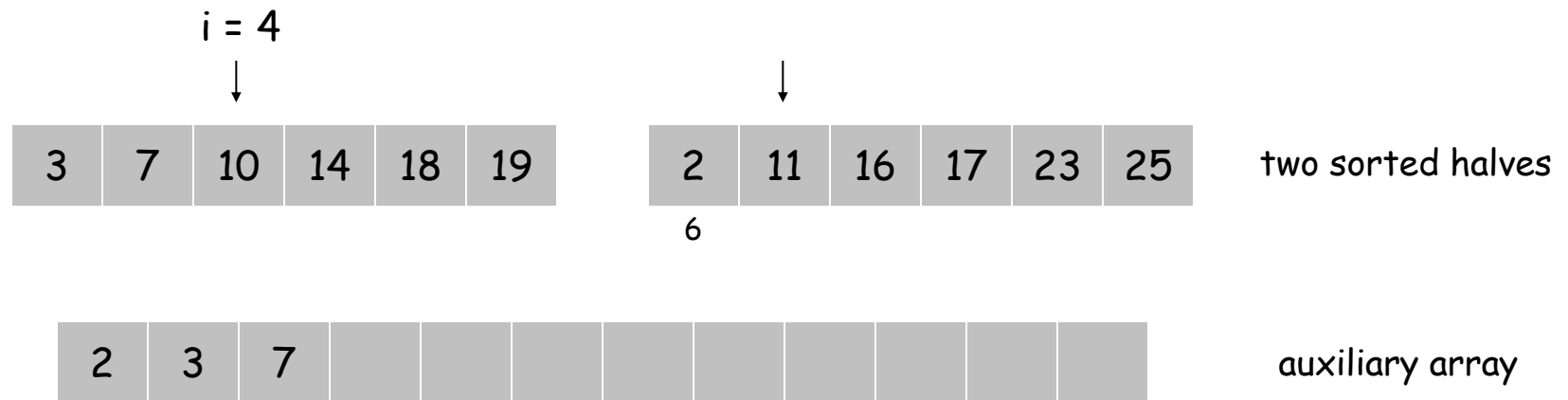


Total: 6

# Merge and Count

## Merge and count step.

- Given two sorted halves, count number of inversions where  $a_i$  and  $a_j$  are in different halves.
- Combine two sorted halves into sorted whole.

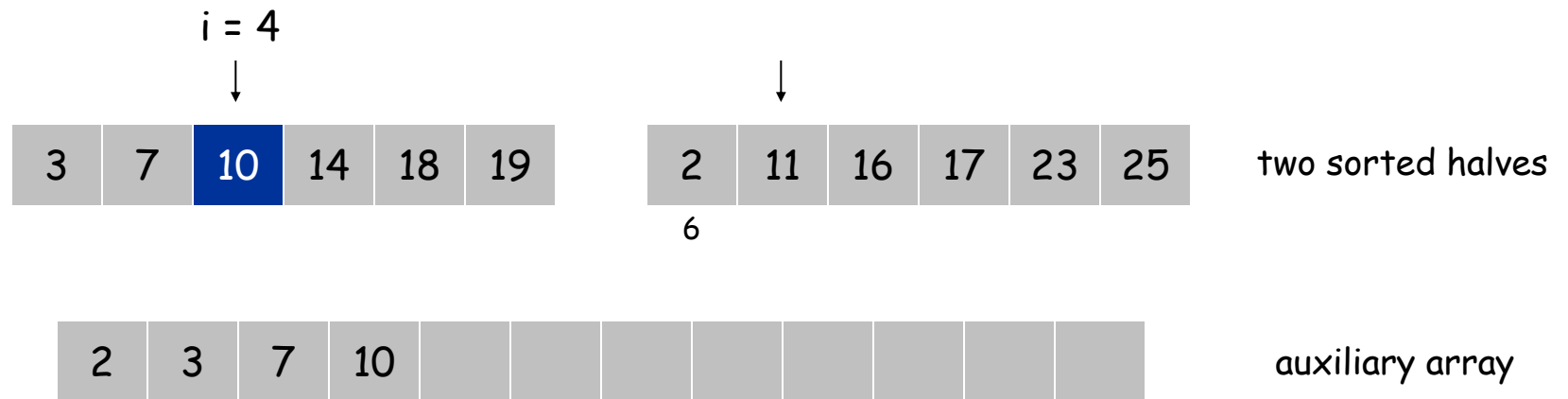


Total: 6

# Merge and Count

## Merge and count step.

- Given two sorted halves, count number of inversions where  $a_i$  and  $a_j$  are in different halves.
- Combine two sorted halves into sorted whole.

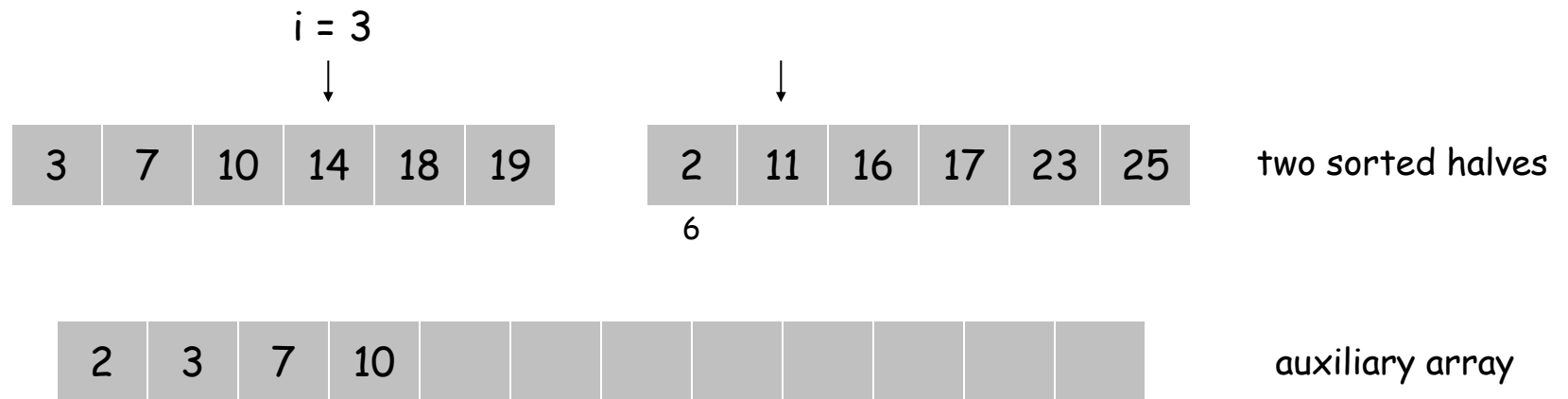


Total: 6

# Merge and Count

## Merge and count step.

- Given two sorted halves, count number of inversions where  $a_i$  and  $a_j$  are in different halves.
- Combine two sorted halves into sorted whole.

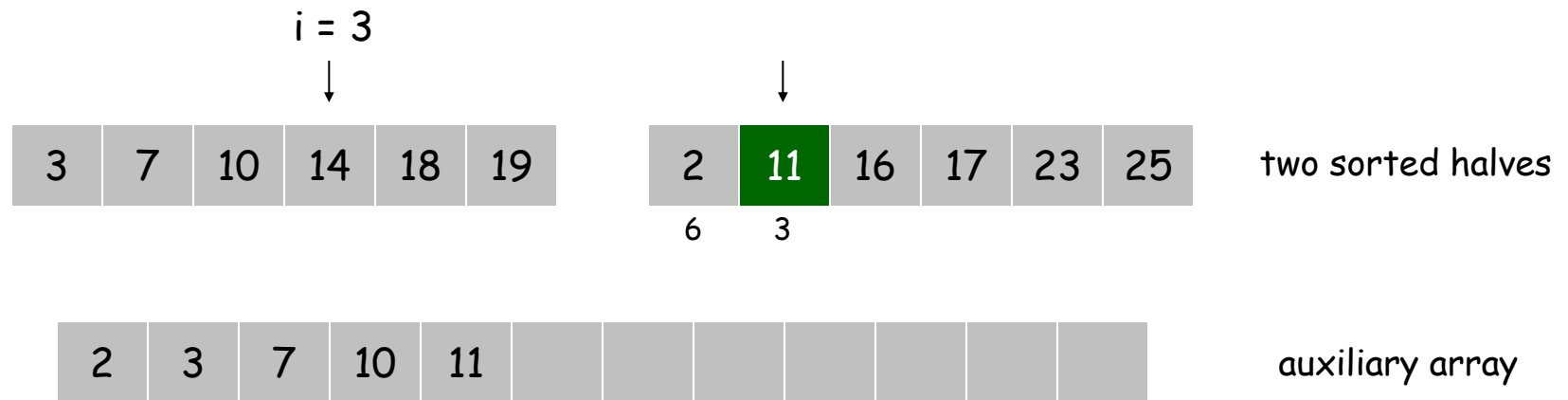


Total: 6

# Merge and Count

## Merge and count step.

- Given two sorted halves, count number of inversions where  $a_i$  and  $a_j$  are in different halves.
- Combine two sorted halves into sorted whole.

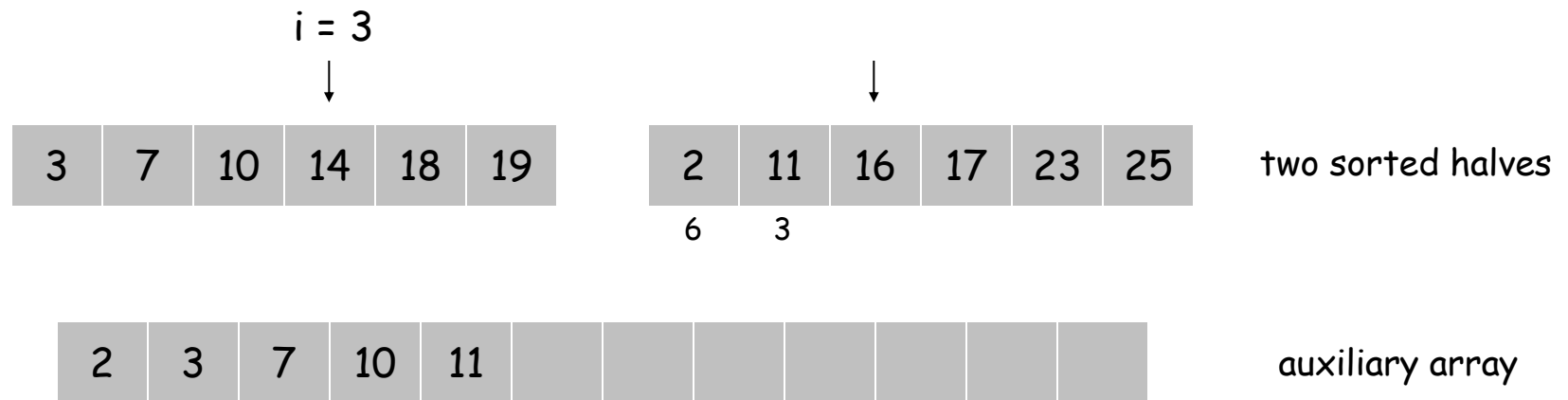


Total: 6 + 3

# Merge and Count

## Merge and count step.

- Given two sorted halves, count number of inversions where  $a_i$  and  $a_j$  are in different halves.
- Combine two sorted halves into sorted whole.

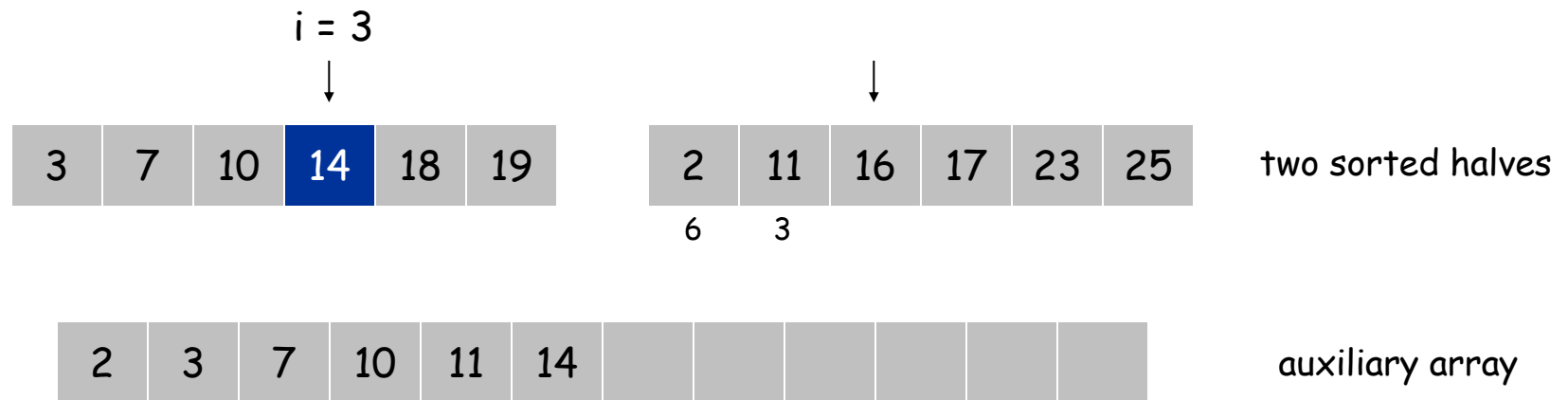


Total: 6 + 3

# Merge and Count

## Merge and count step.

- Given two sorted halves, count number of inversions where  $a_i$  and  $a_j$  are in different halves.
- Combine two sorted halves into sorted whole.

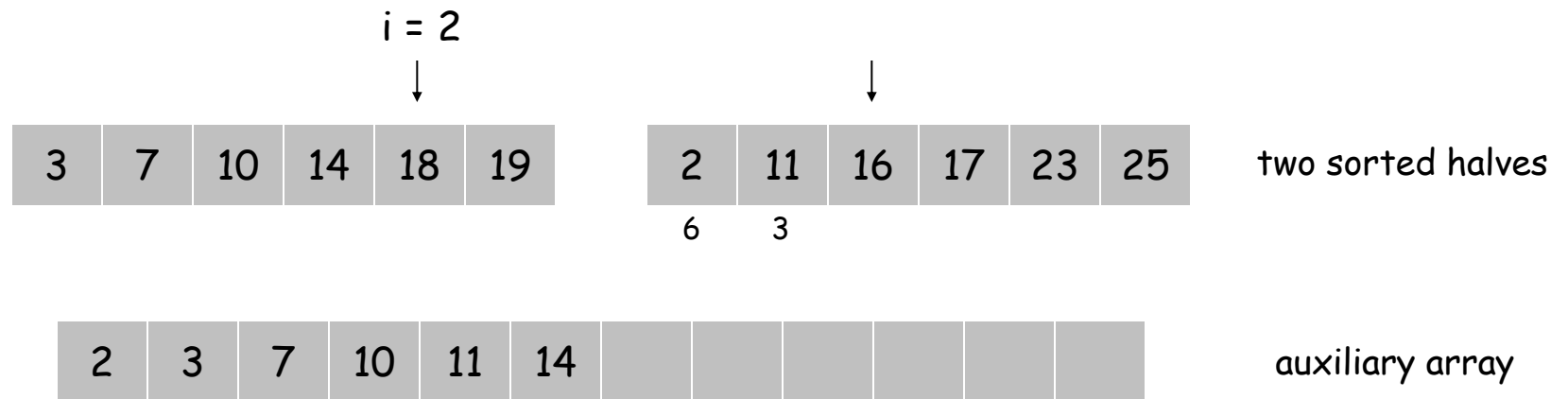


Total: 6 + 3

# Merge and Count

## Merge and count step.

- Given two sorted halves, count number of inversions where  $a_i$  and  $a_j$  are in different halves.
- Combine two sorted halves into sorted whole.



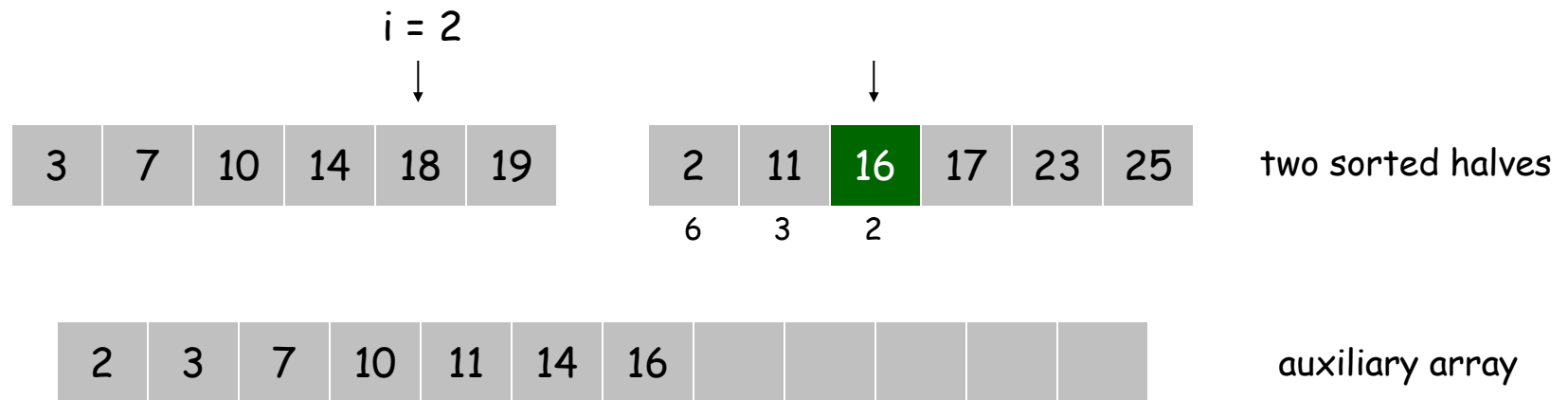
Total: 6 + 3



# Merge and Count

## Merge and count step.

- Given two sorted halves, count number of inversions where  $a_i$  and  $a_j$  are in different halves.
- Combine two sorted halves into sorted whole.

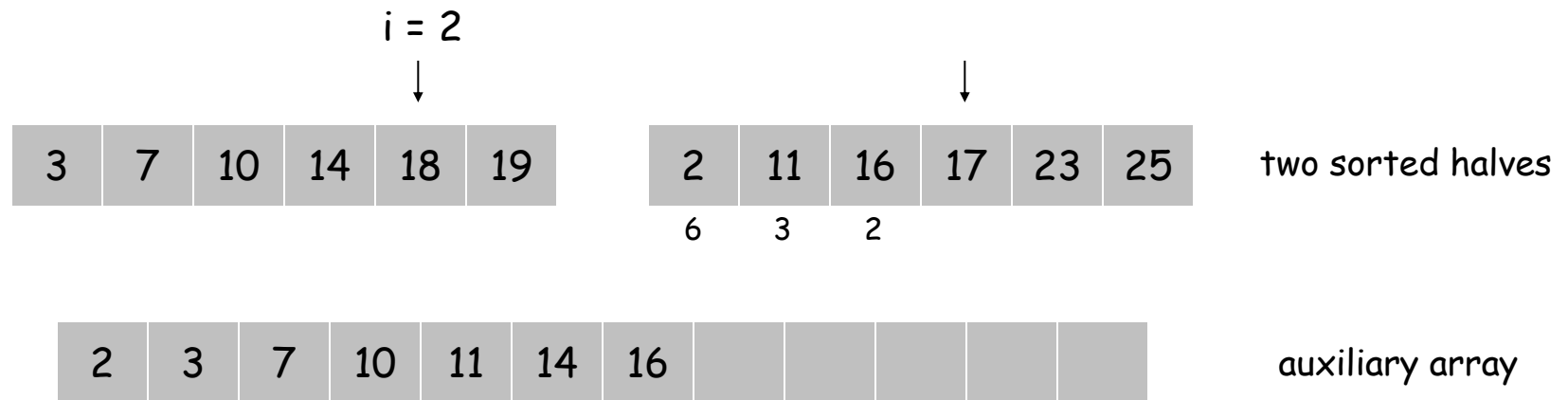


Total:  $6 + 3 + 2$

# Merge and Count

## Merge and count step.

- Given two sorted halves, count number of inversions where  $a_i$  and  $a_j$  are in different halves.
- Combine two sorted halves into sorted whole.

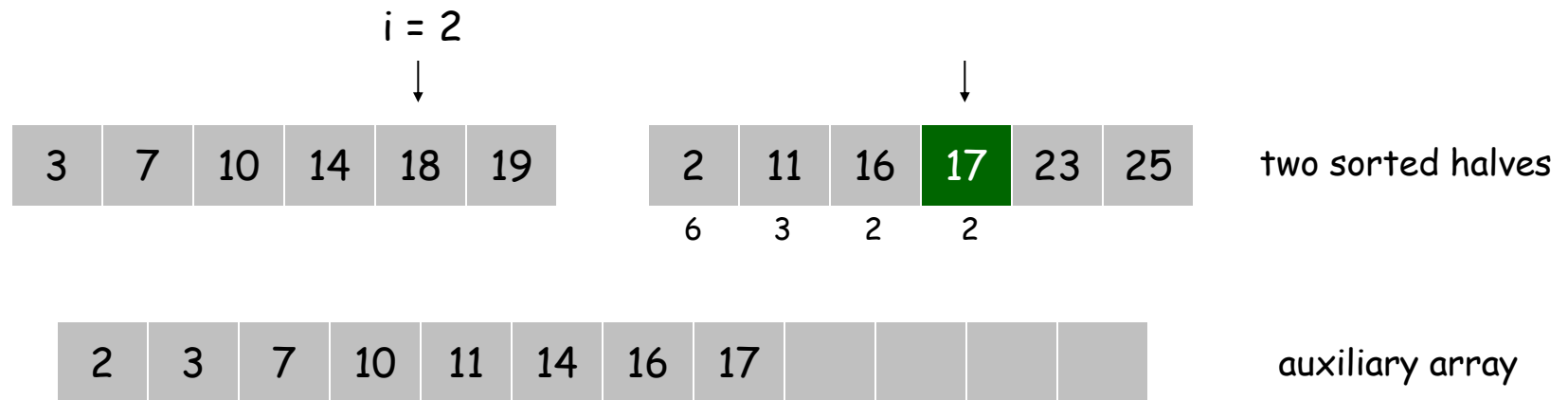


Total:  $6 + 3 + 2$

# Merge and Count

## Merge and count step.

- Given two sorted halves, count number of inversions where  $a_i$  and  $a_j$  are in different halves.
- Combine two sorted halves into sorted whole.

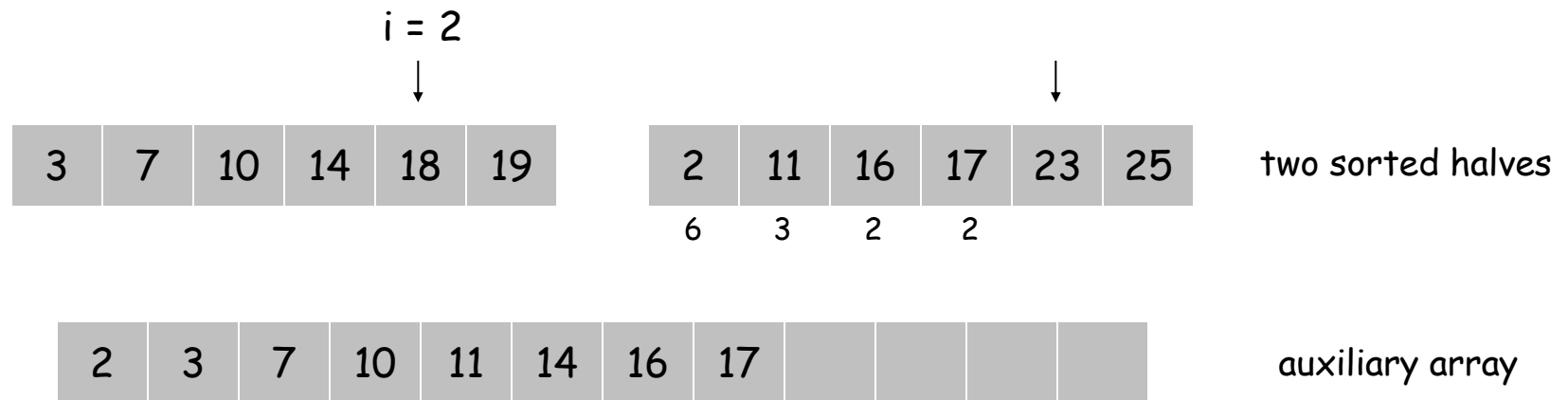


Total:  $6 + 3 + 2 + 2$

# Merge and Count

## Merge and count step.

- Given two sorted halves, count number of inversions where  $a_i$  and  $a_j$  are in different halves.
- Combine two sorted halves into sorted whole.

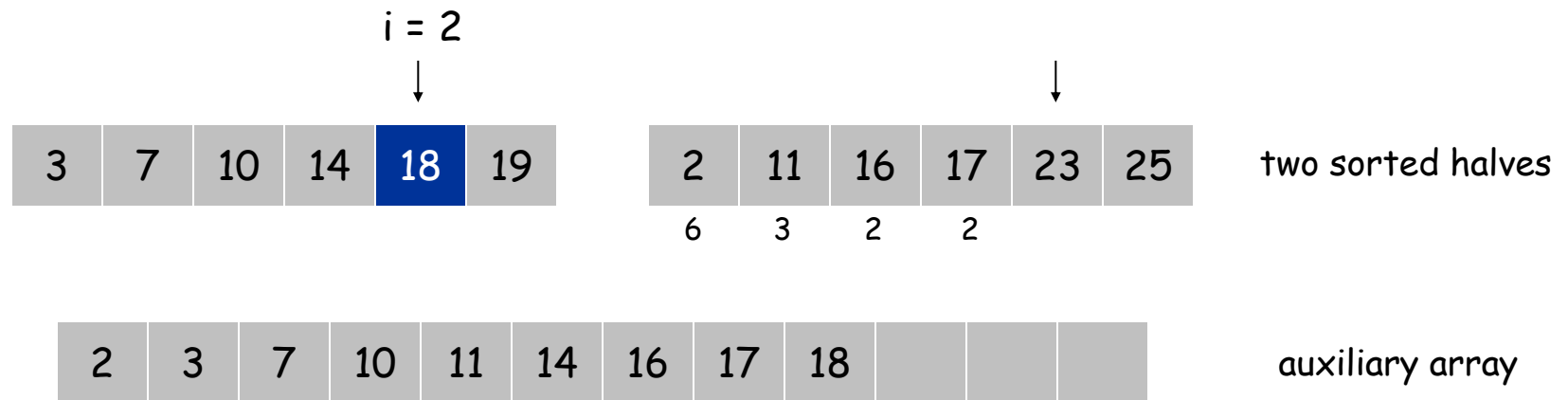


Total:  $6 + 3 + 2 + 2$

# Merge and Count

## Merge and count step.

- Given two sorted halves, count number of inversions where  $a_i$  and  $a_j$  are in different halves.
- Combine two sorted halves into sorted whole.

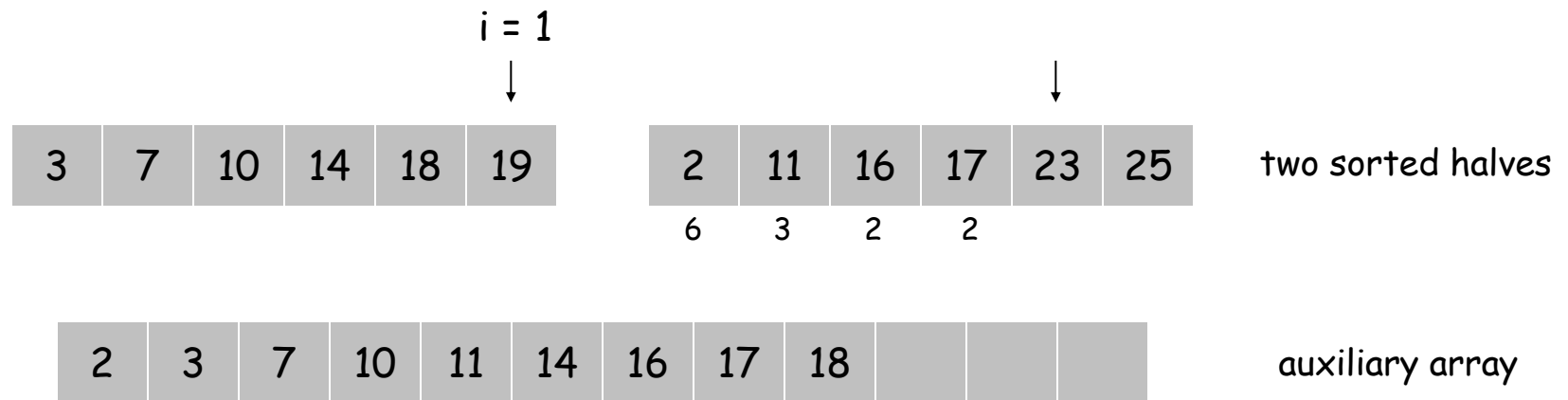


Total:  $6 + 3 + 2 + 2$

# Merge and Count

## Merge and count step.

- Given two sorted halves, count number of inversions where  $a_i$  and  $a_j$  are in different halves.
- Combine two sorted halves into sorted whole.

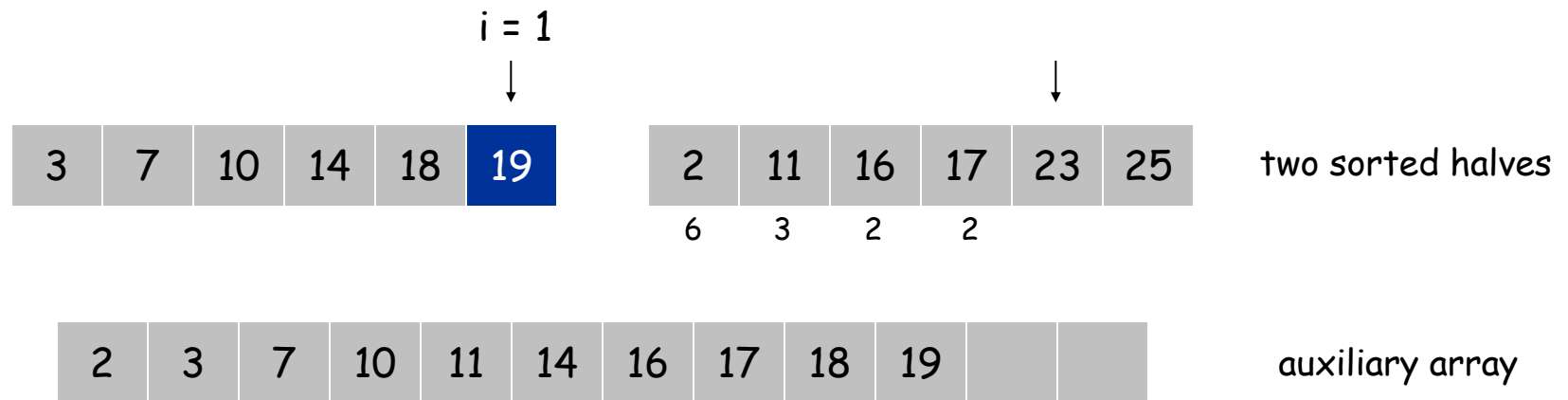


Total:  $6 + 3 + 2 + 2$

# Merge and Count

## Merge and count step.

- Given two sorted halves, count number of inversions where  $a_i$  and  $a_j$  are in different halves.
- Combine two sorted halves into sorted whole.

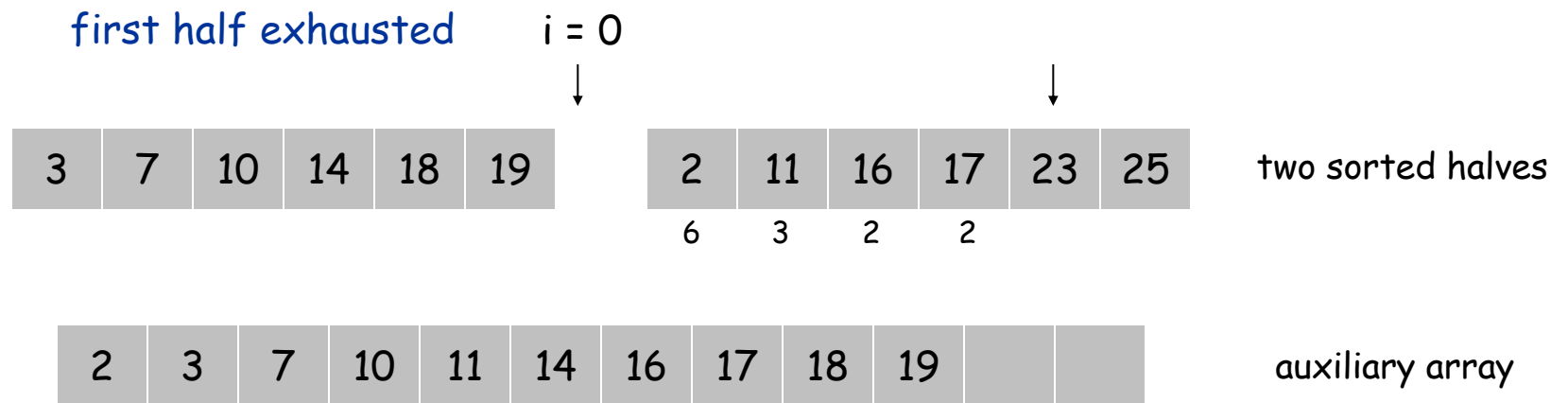


Total:  $6 + 3 + 2 + 2$

# Merge and Count

## Merge and count step.

- Given two sorted halves, count number of inversions where  $a_i$  and  $a_j$  are in different halves.
- Combine two sorted halves into sorted whole.



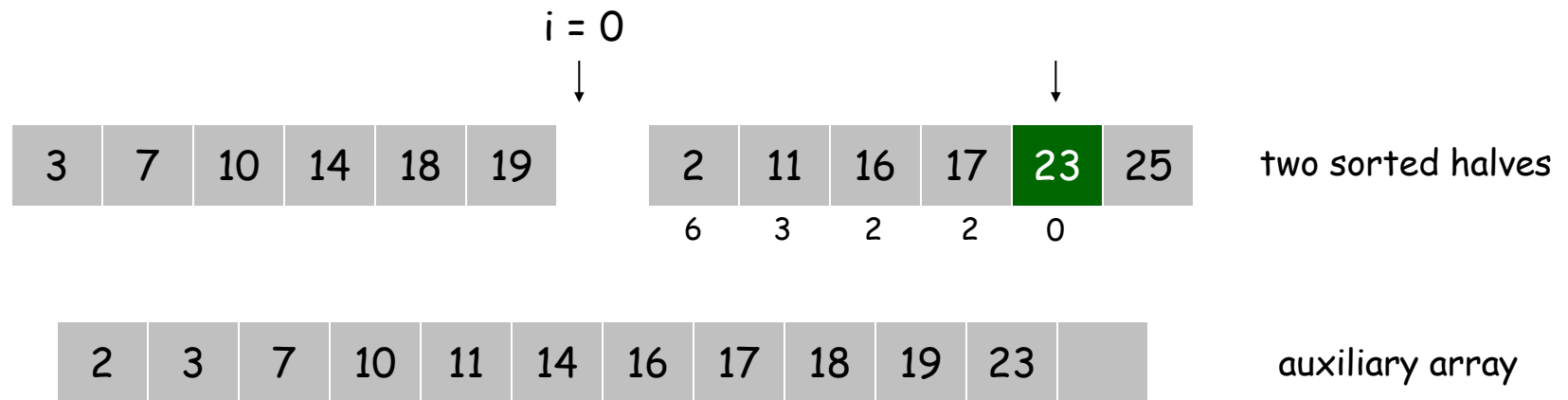
Total:  $6 + 3 + 2 + 2$



# Merge and Count

## Merge and count step.

- Given two sorted halves, count number of inversions where  $a_i$  and  $a_j$  are in different halves.
- Combine two sorted halves into sorted whole.

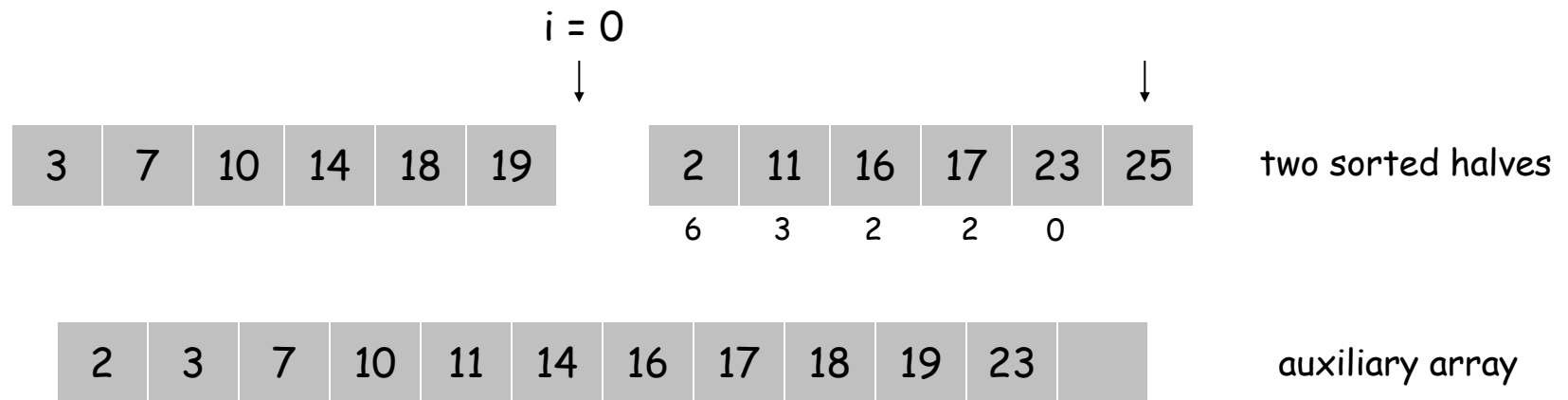


Total:  $6 + 3 + 2 + 2 + 0$

# Merge and Count

## Merge and count step.

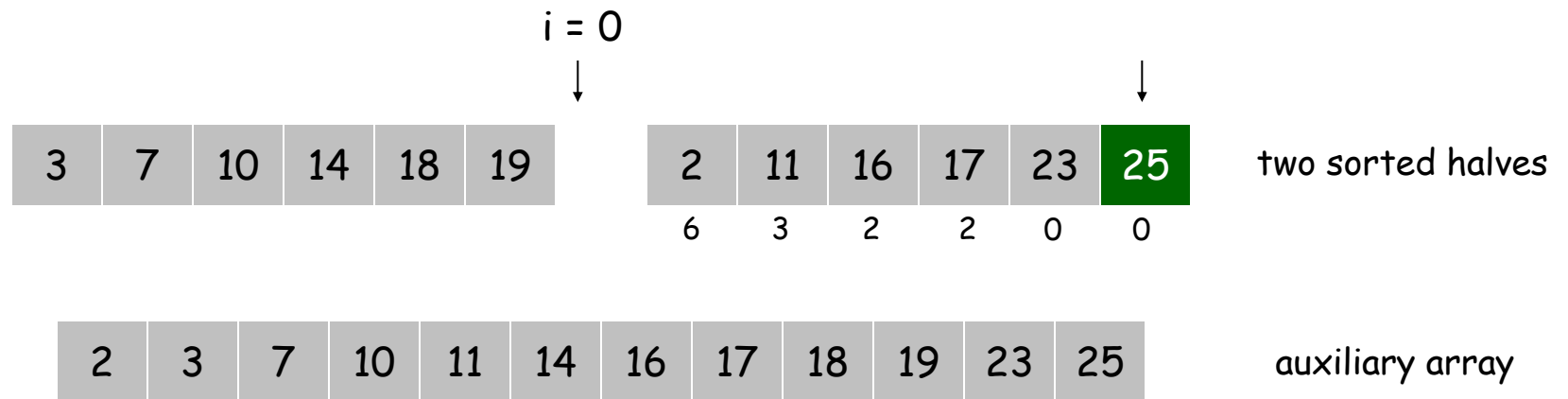
- Given two sorted halves, count number of inversions where  $a_i$  and  $a_j$  are in different halves.
- Combine two sorted halves into sorted whole.



# Merge and Count

## Merge and count step.

- Given two sorted halves, count number of inversions where  $a_i$  and  $a_j$  are in different halves.
- Combine two sorted halves into sorted whole.

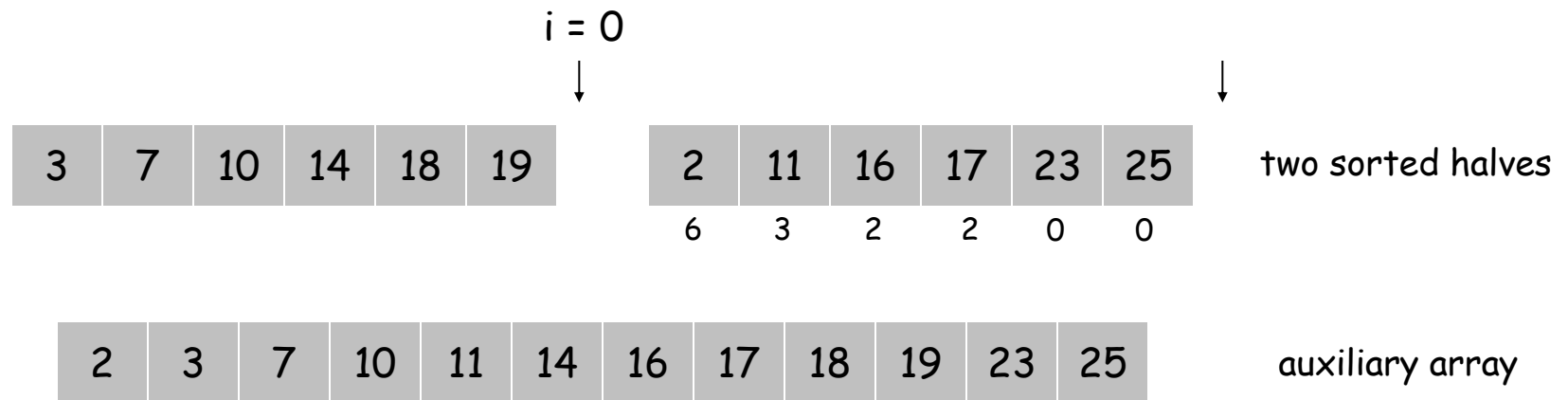


Total:  $6 + 3 + 2 + 2 + 0 + 0$

# Merge and Count

## Merge and count step.

- Given two sorted halves, count number of inversions where  $a_i$  and  $a_j$  are in different halves.
- Combine two sorted halves into sorted whole.



Total:  $6 + 3 + 2 + 2 + 0 + 0 = 13$