UNIVERSITY OF CALIFORNIA
UC RIVERSIDE

Sorting

Chapter 7

Heap Sort

- Make use of log(n) insertion and deletion in min-heaps
- Insert all elements into a min-heap
- Remove all elements from the min-heap

```
Initialize H as a MinHeap
For i = 1 to n
  Insert A(i) into H
For i = 1 to n
  A[i] = DeleteMin from H
```

$$T(n) = O(n \log n)$$

Heap Sort Example

A=	8	10	5	1	3	20	13	7	2	12
H=										

Heap Sort Example

Phase I: Building the Heap

A=	8	10	5	1	3	20	13	7	2	12

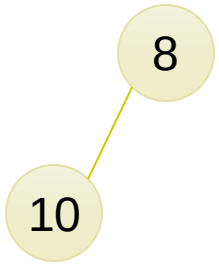
H=	8									

8

Heap Sort Example

Phase I: Building the Heap

A=	8	10	5	1	3	20	13	7	2	12
		↑								
H=	8	10								

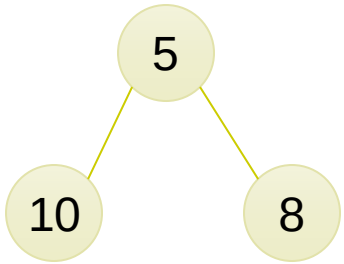


Heap Sort Example

Phase I: Building the Heap

A=	8	10	5	1	3	20	13	7	2	12

H=	5	10	8							

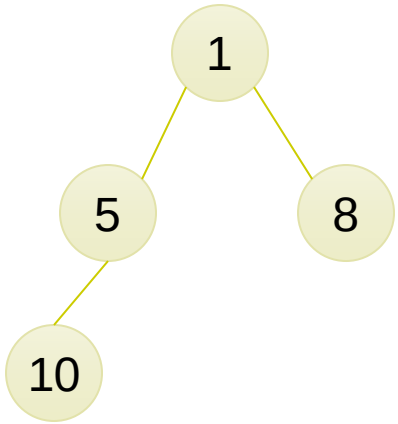


Heap Sort Example

Phase I: Building the Heap

A=	8	10	5	1	3	20	13	7	2	12

H=	1	5	8	10						

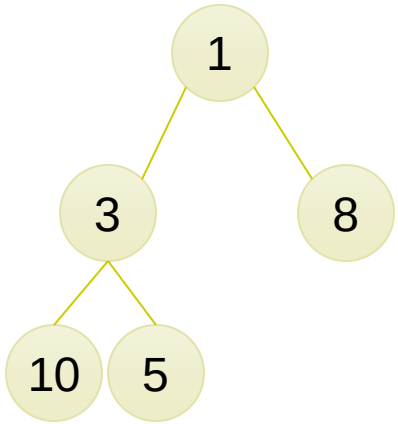


Heap Sort Example

Phase I: Building the Heap

A=	8	10	5	1	3	20	13	7	2	12

H=	1	3	8	10	5					

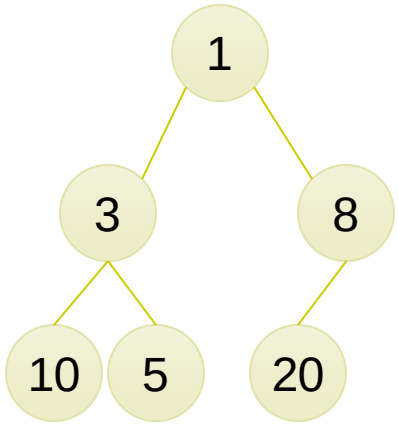


Heap Sort Example

Phase I: Building the Heap

A=	8	10	5	1	3	20	13	7	2	12

H=	1	3	8	10	5	20				

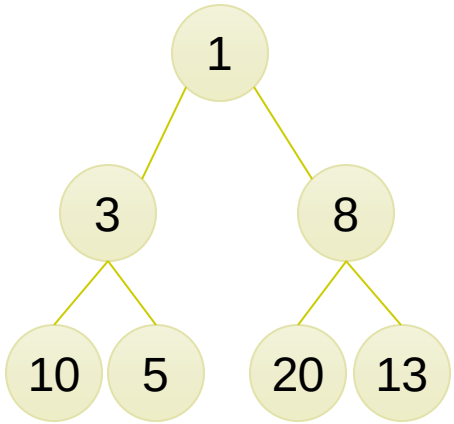


Heap Sort Example

Phase I: Building the Heap

A=	8	10	5	1	3	20	13	7	2	12

H=	1	3	8	10	5	20	13			



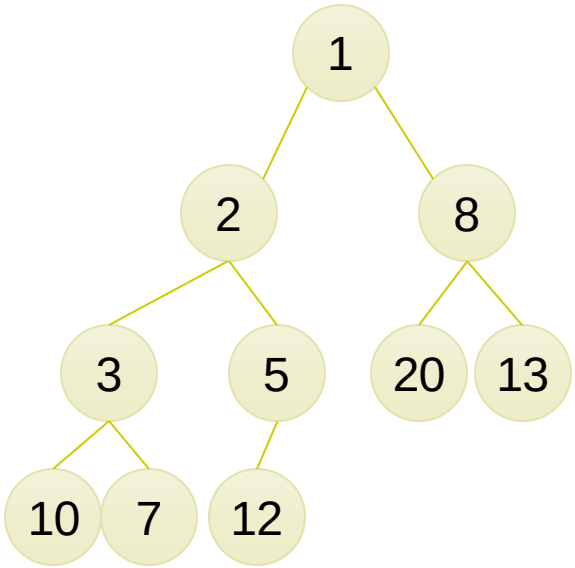
Heap Sort Example

Phase I: Building the Heap

A=	8	10	5	1	3	20	13	7	2	12

↑

H=	1	2	8	3	5	20	13	10	7	12

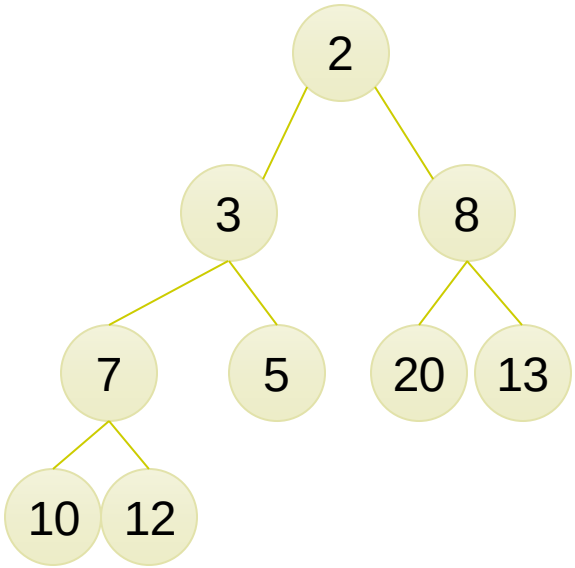


Heap Sort Example

Phase II: Deleting from the Heap

A=	1	10	5	1	3	20	13	7	2	12

H=	2	3	8	7	5	20	13	10	12	

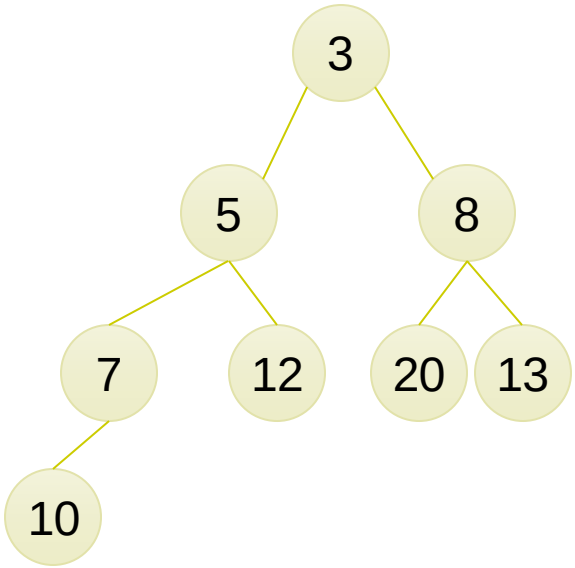


Heap Sort Example

Phase II: Deleting from the Heap

A=	1	2	5	1	3	20	13	7	2	12

H=	3	5	8	7	12	20	13	10		

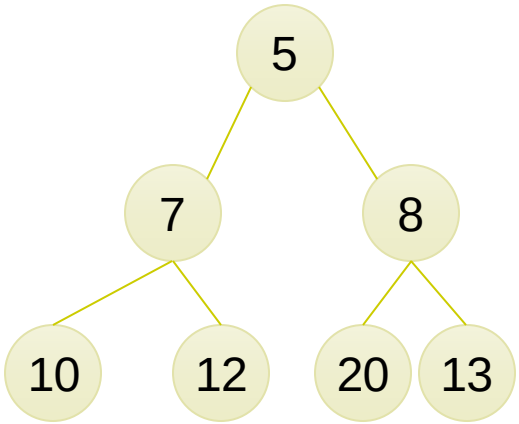


Heap Sort Example

Phase II: Deleting from the Heap

A=	1	2	3	1	3	20	13	7	2	12

H=	5	7	8	10	12	20	13			



Heap Sort Example

Phase II: Deleting from the Heap

A=	1	2	3	5	7	8	10	12	13	20

H=										

In-place Heap Sort

- Reuse the same array A as a Heap

```
For i = 2 to n
  Insert A[i] into the MaxHeap A[1..i-1]
For i = n downto 1
  A[i] = DeleteMax from Heap A[1..i]
```

In-place Heap Sort Example

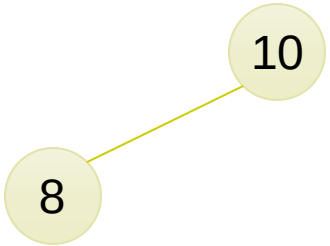
A=	8	10	5	1	3	20	13	7	2	12

In-place Heap Sort

Phase I: Building the MaxHeap

A=

10	8	5	1	3	20	13	7	2	12

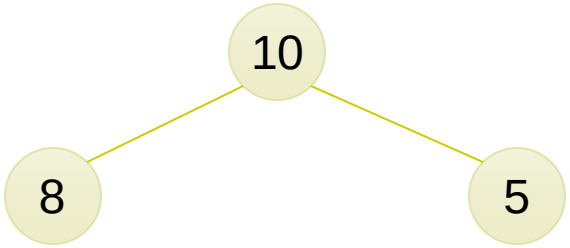


In-place Heap Sort

Phase I: Building the MaxHeap

A=

10	8	5	1	3	20	13	7	2	12

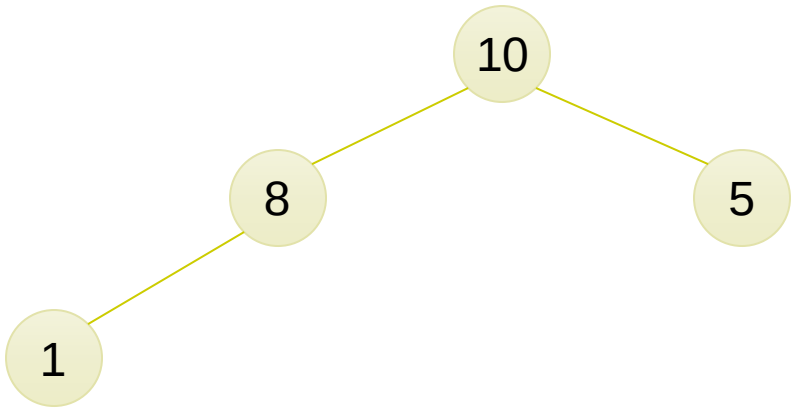


In-place Heap Sort

Phase I: Building the MaxHeap

A=

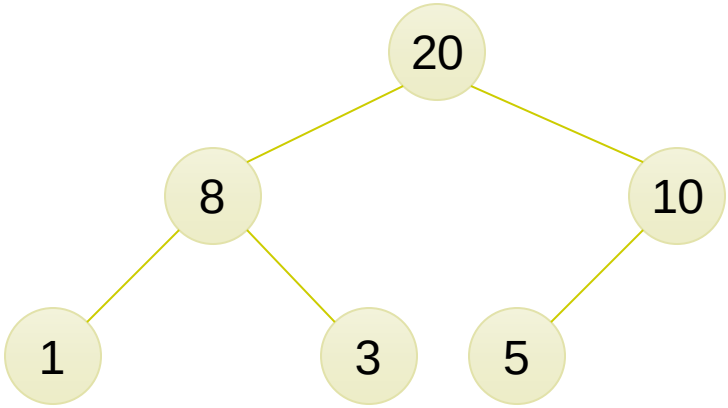
10	8	5	1	3	20	13	7	2	12



In-place Heap Sort

Phase I: Building the MaxHeap

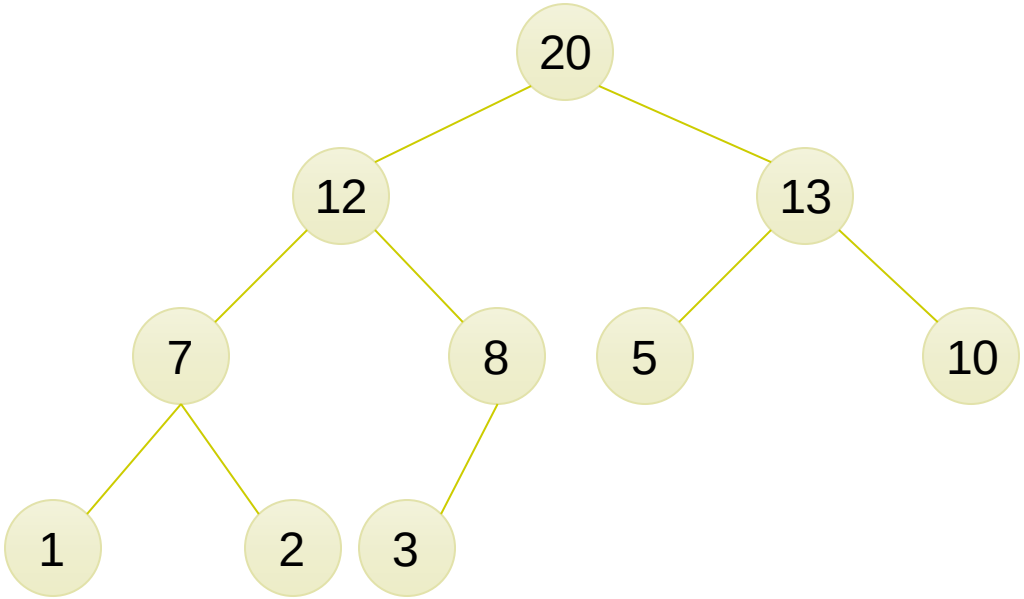
A=	20	8	10	1	3	5	13	7	2	12



In-place Heap Sort

Phase I: Building the MaxHeap

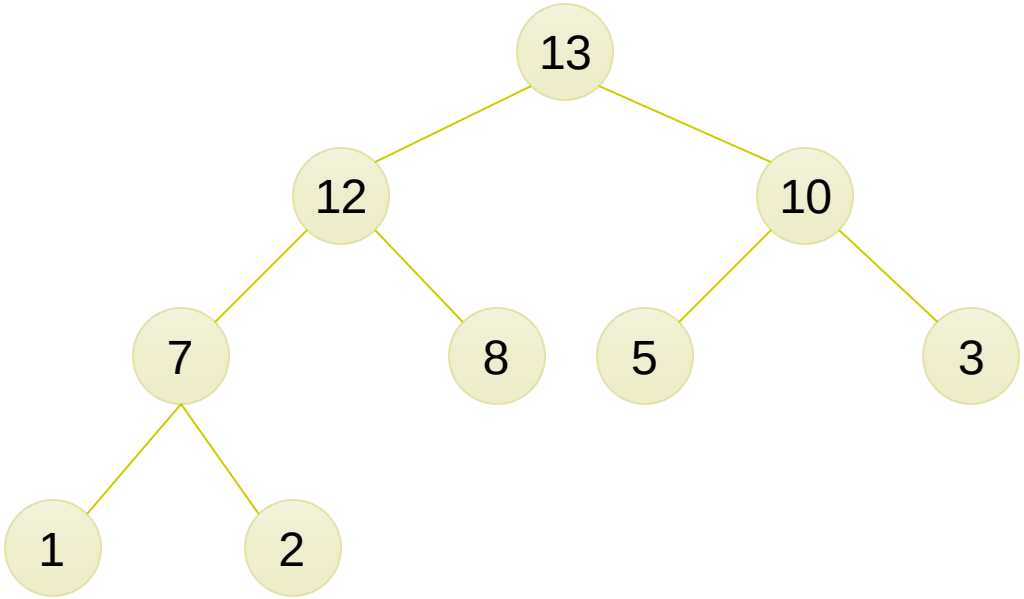
A=	20	12	13	7	8	5	10	1	2	3



In-place Heap Sort

Phase II: Deleting from the MaxHeap

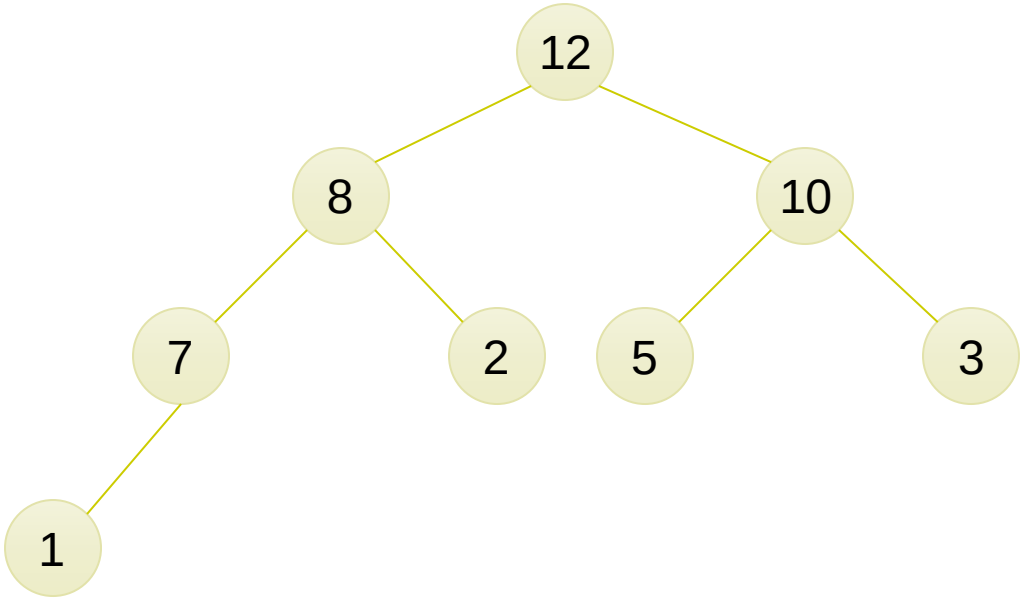
A=	13	12	10	7	8	5	3	1	2	20



In-place Heap Sort

Phase II: Deleting from the MaxHeap

A=	12	8	10	7	2	5	3	1	13	20

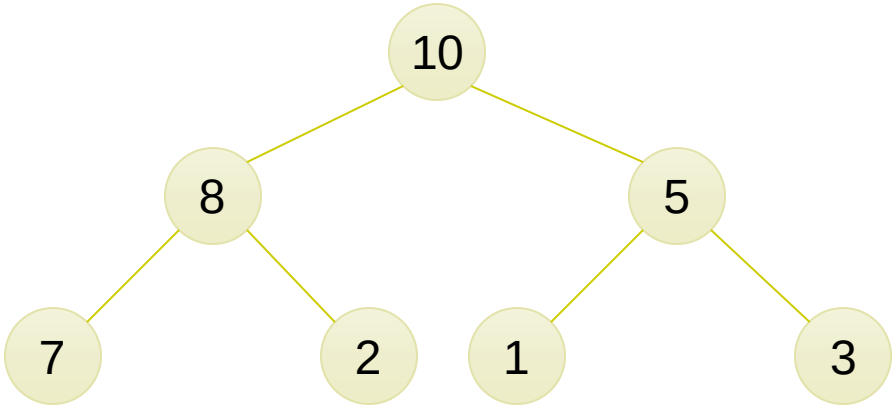


In-place Heap Sort

Phase II: Deleting from the MaxHeap

A=

10	8	5	7	2	1	3	12	13	20

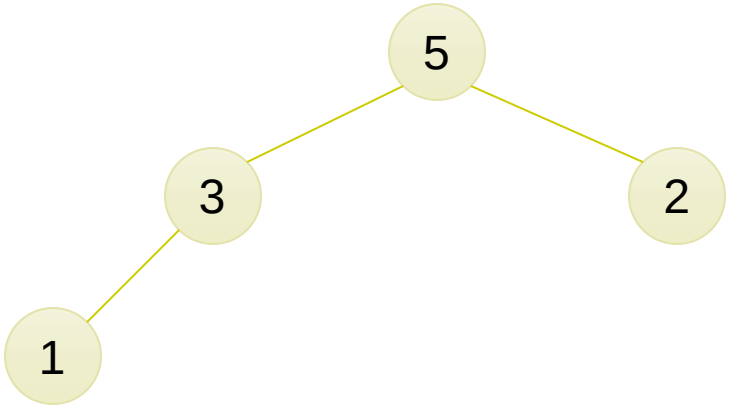


In-place Heap Sort

Phase II: Deleting from the MaxHeap

A=

5	3	2	1	7	8	10	12	13	20



In-place Heap Sort

Phase II: Deleting from the MaxHeap

A=

1	2	3	5	7	8	10	12	13	20

↑

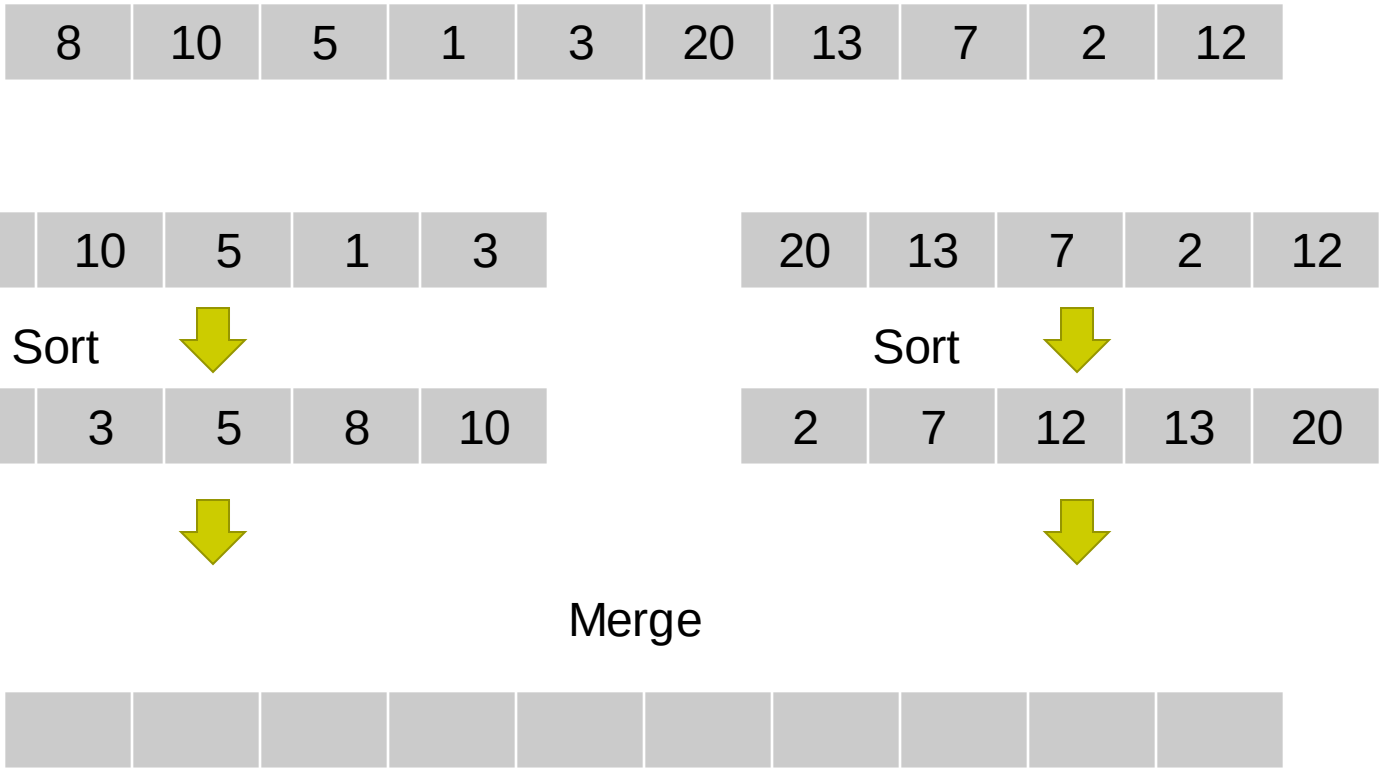
Merge-sort

- Split the array into two subarrays
- Sort each subarray
- Merge the two sorted arrays

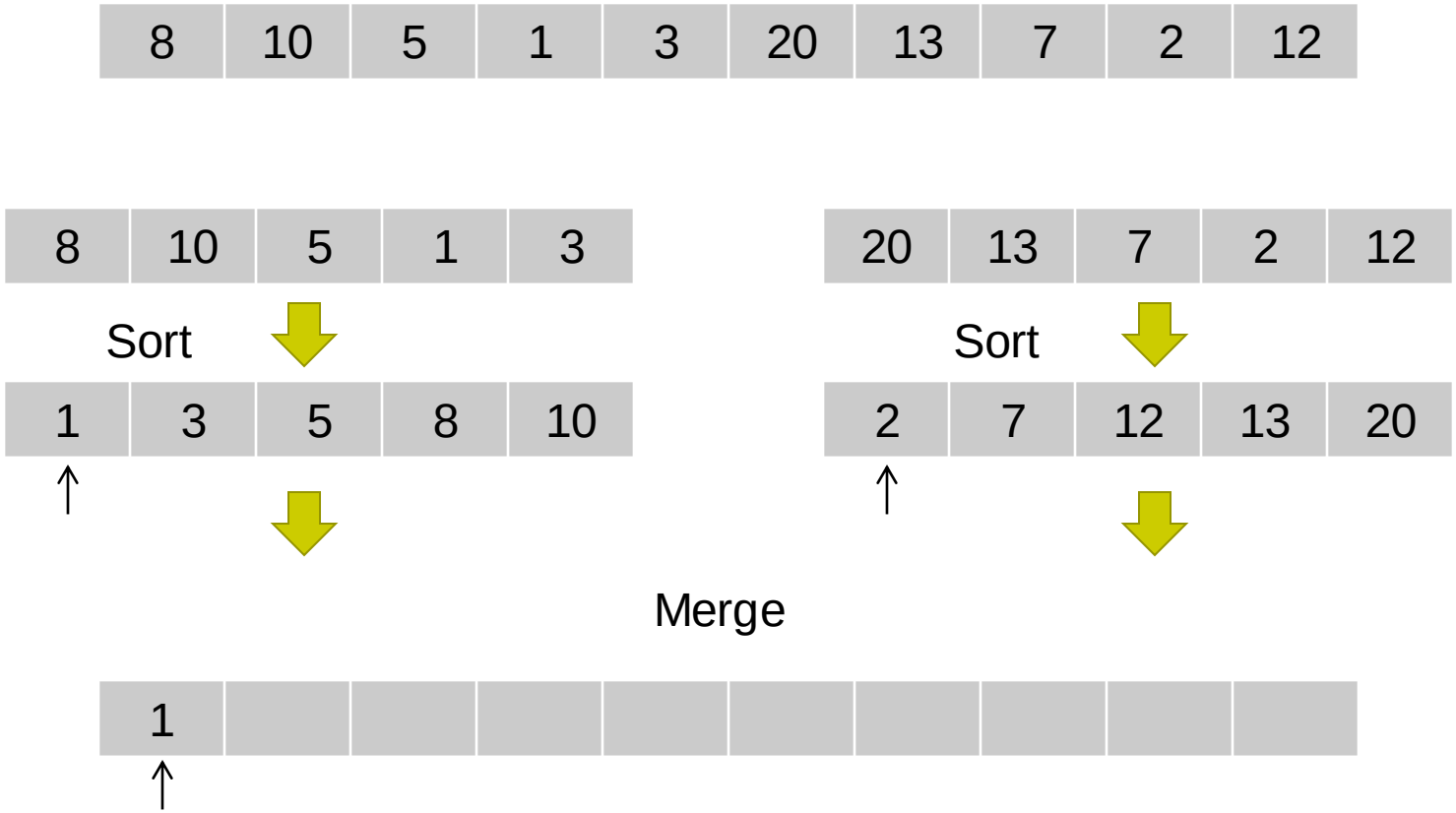
```
Sort(A, start, end)
  return if start ≥ end
  m = ⌊n/2⌋
  Sort(A, start, m)
  Sort(A, m+1, end)
  Merge(A, start, m+1, end)

Sort(A, n)
  Sort(A, 1, n)
```

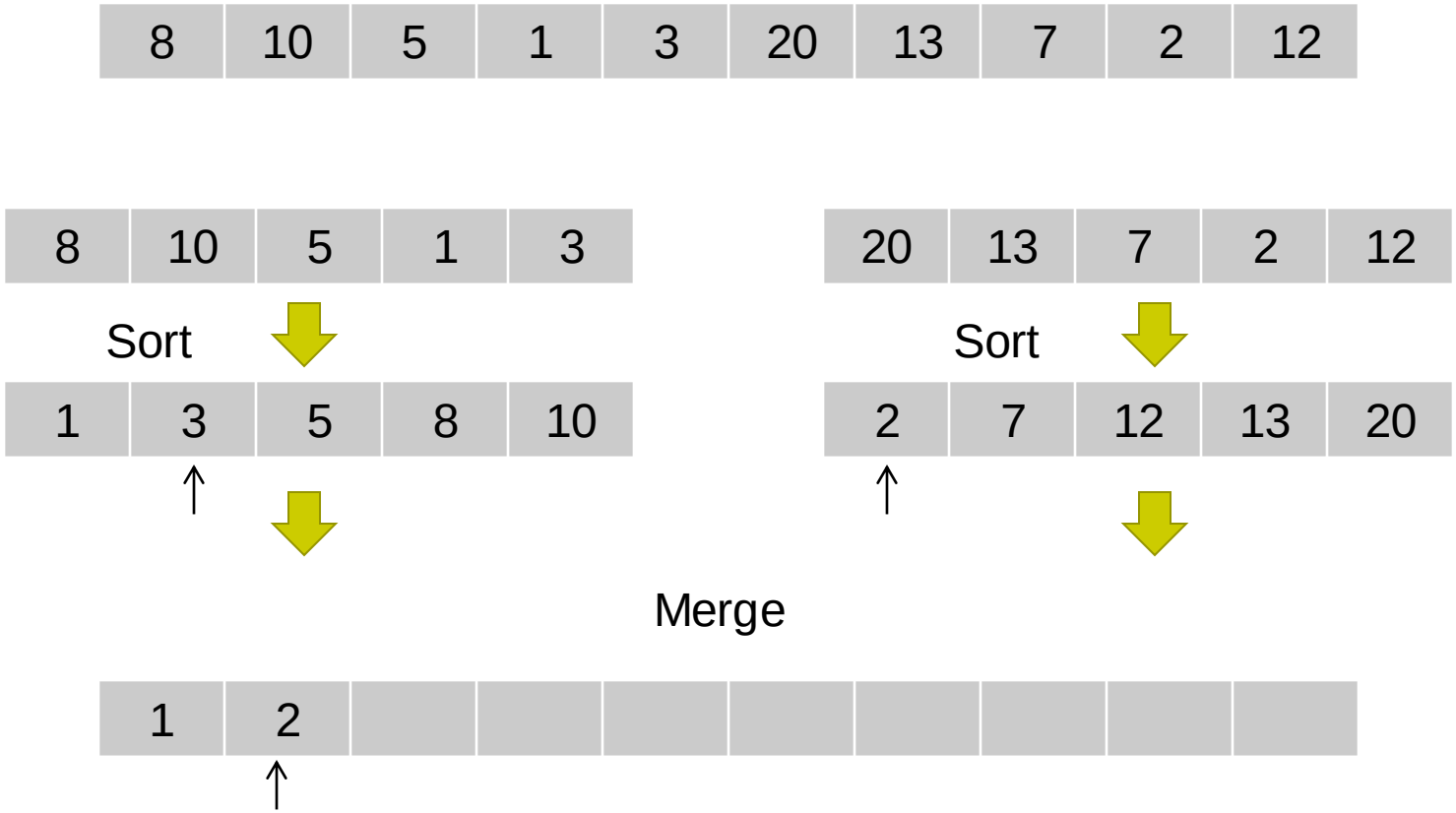
Merging Example



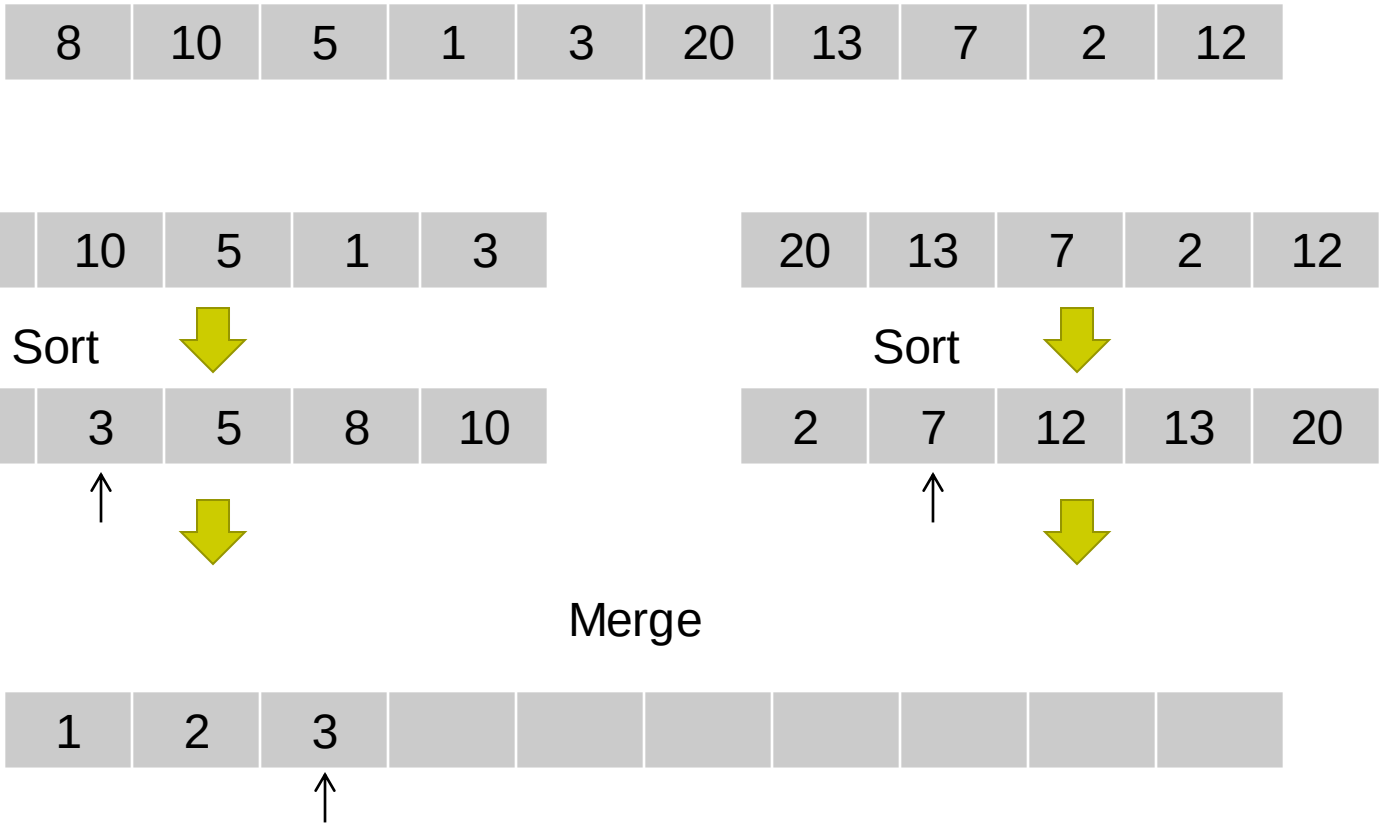
Merging Example



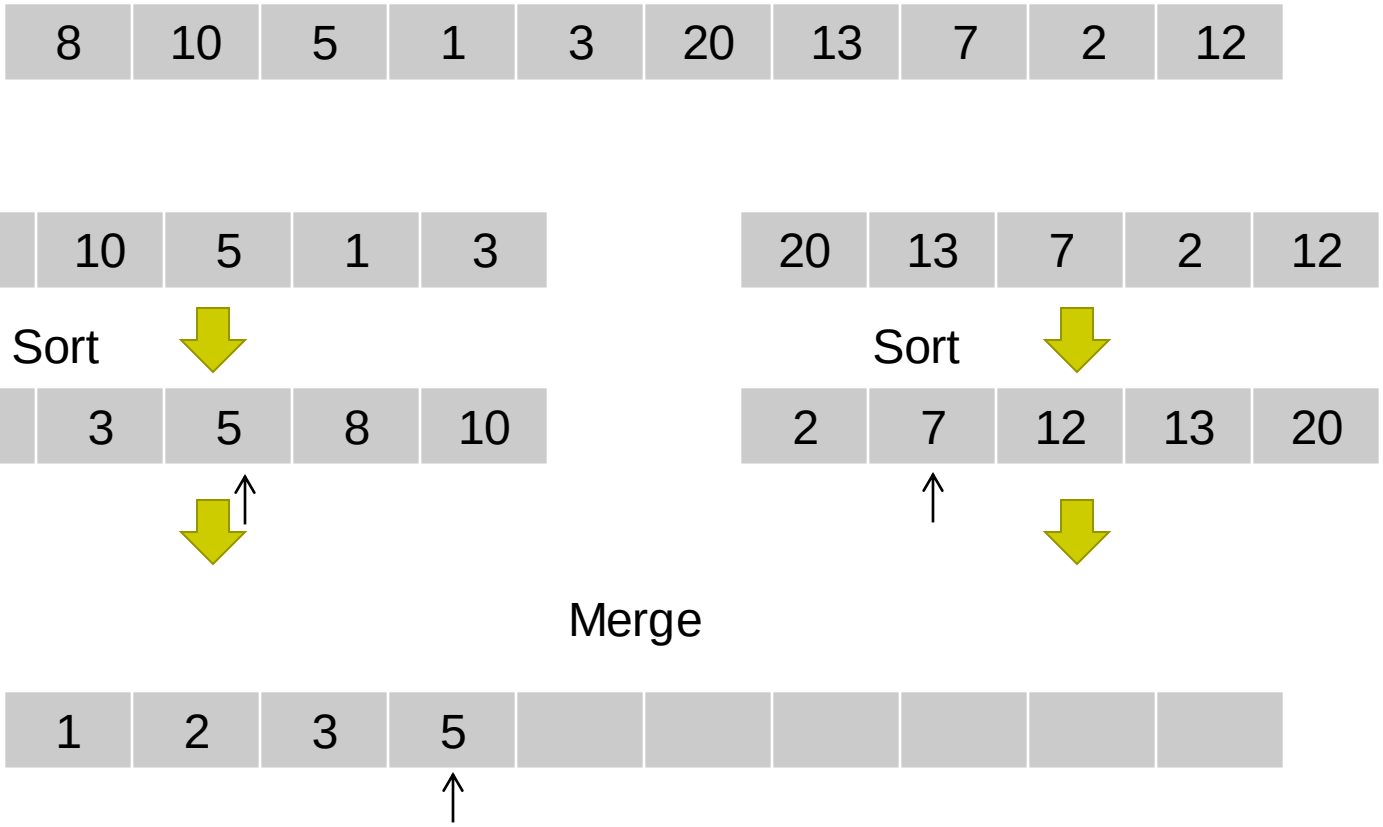
Merging Example



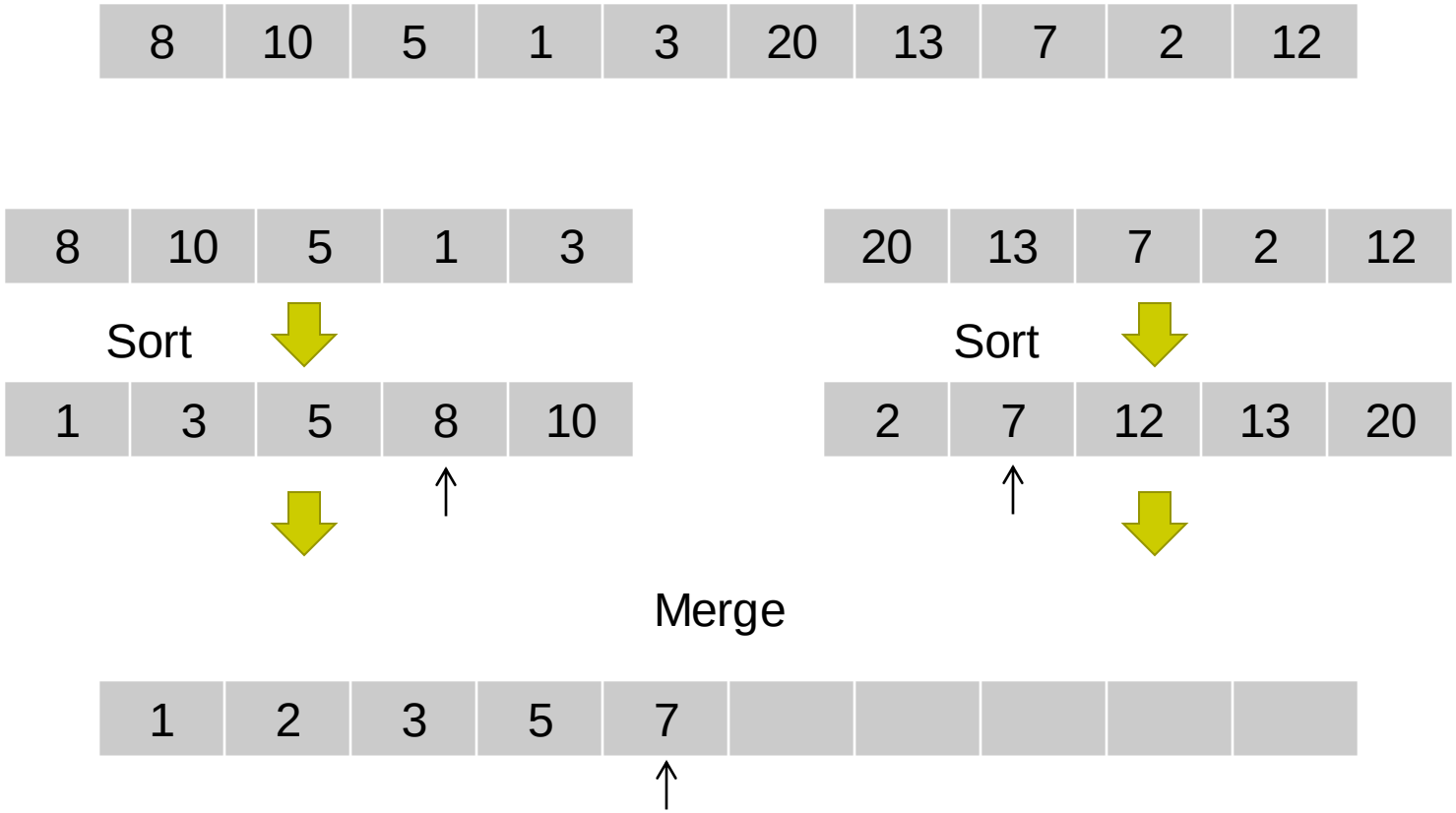
Merging Example



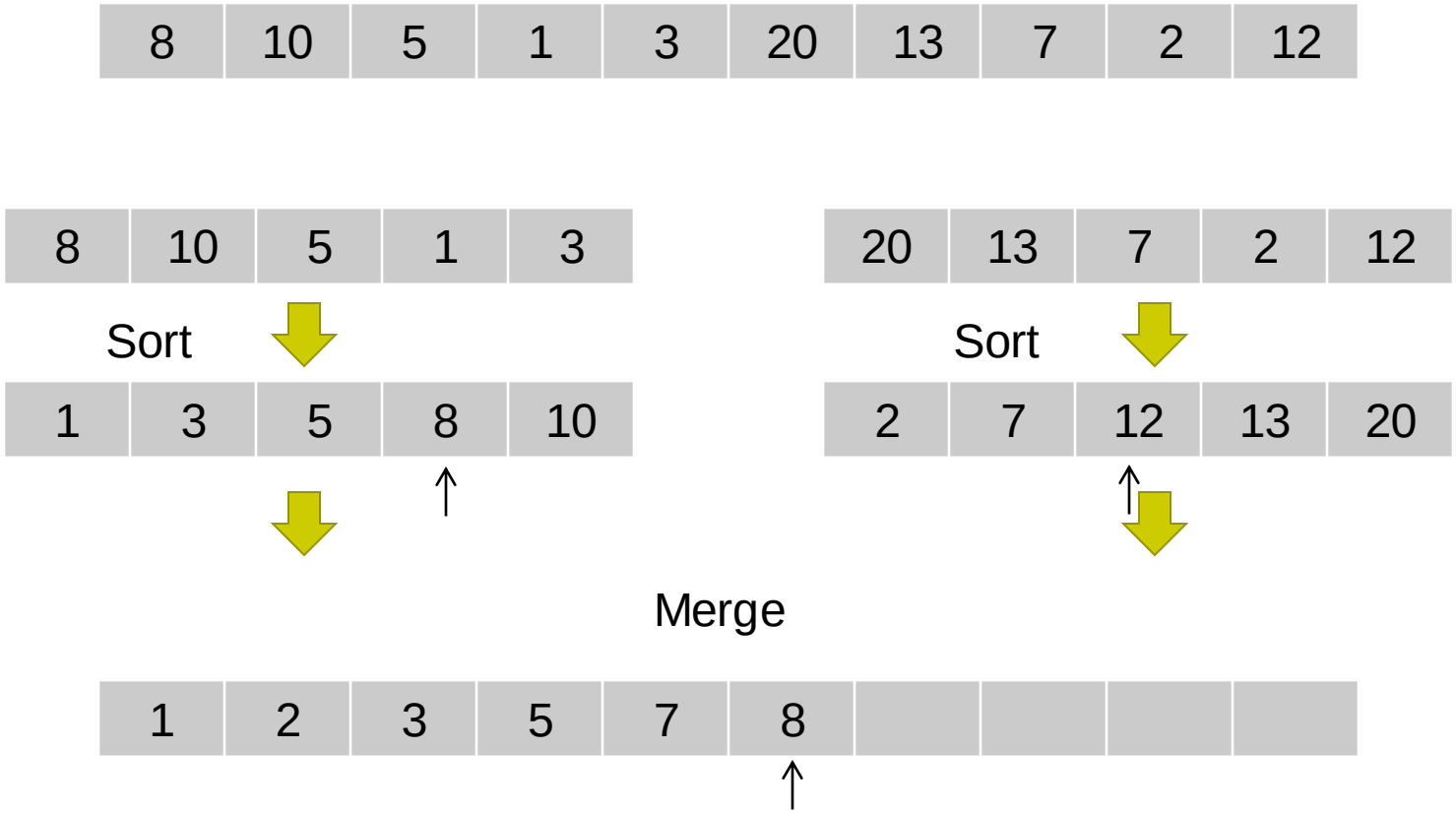
Merging Example



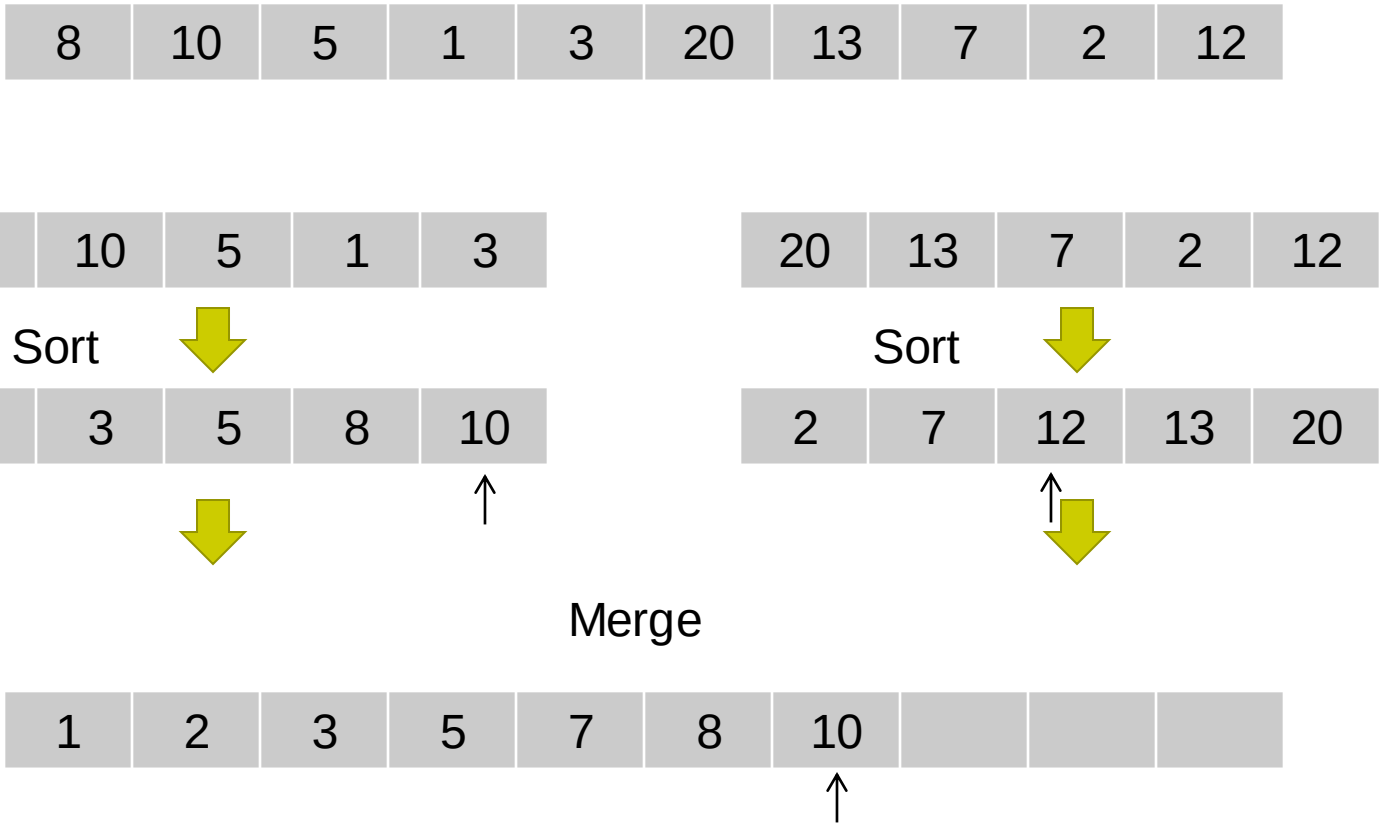
Merging Example



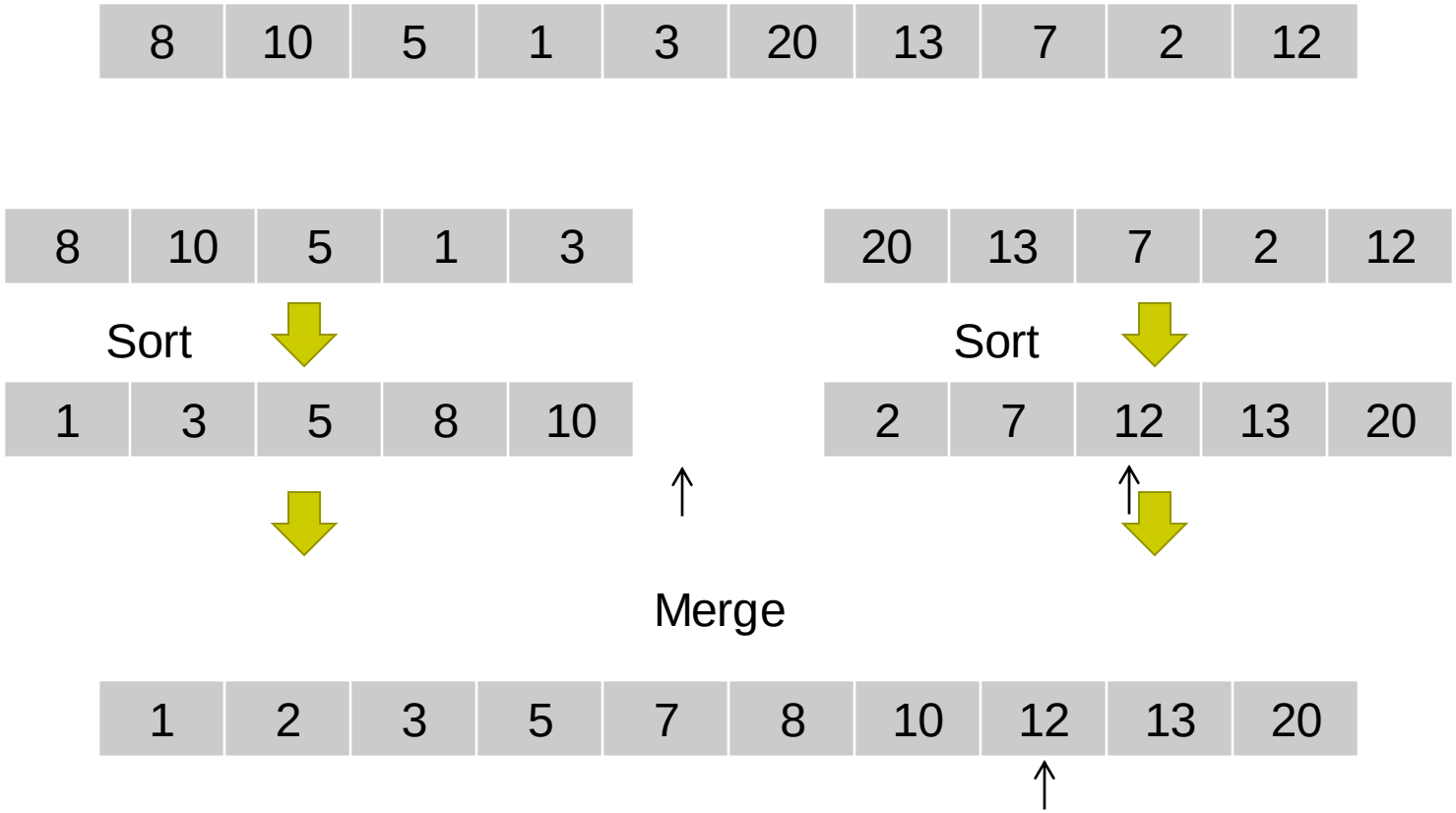
Merging Example



Merging Example



Merging Example



Merge

```
Merge(A, start, m, end)
  B = new array of size end – start + 1
  i = start
  j = m+1
  k = 1
  while i ≤ m AND j ≤ end
    if A[i] < A[j]
      B[k++] = A[i++]
    else
      B[k++] = A[j++]
  while i ≤ m
    B[k++] = A[i++]
  while j ≤ end
    B[k++] = A[j++]
  Copy B back into A[start..end]
```

Merge-sort Example

8	10	5	1	3	20	13	7	2	12

8	10	5	1	3

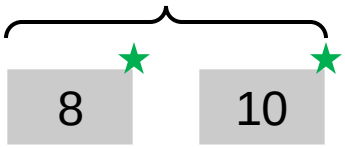
20	13	7	2	12

8	10	5

1	3

8	10

5



8	10

Merge-sort Example

8	10	5	1	3	20	13	7	2	12

8	10	5	1	3

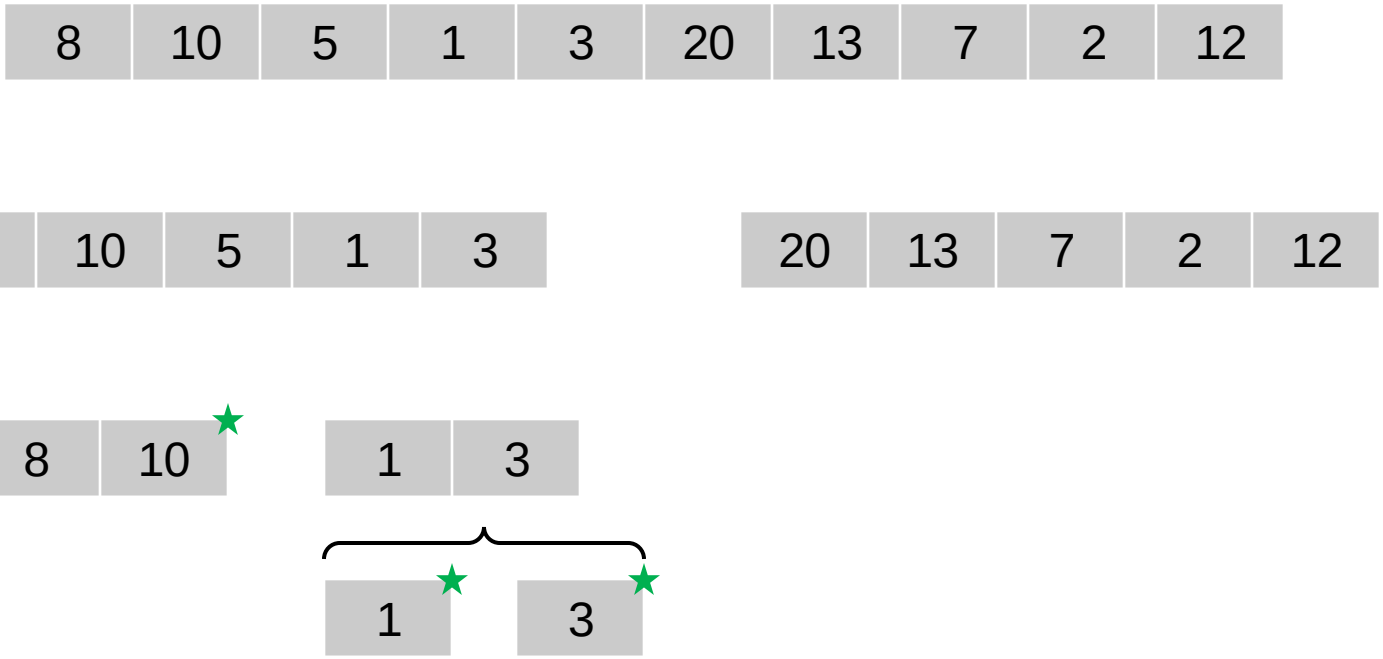
20	13	7	2	12

8	10	5

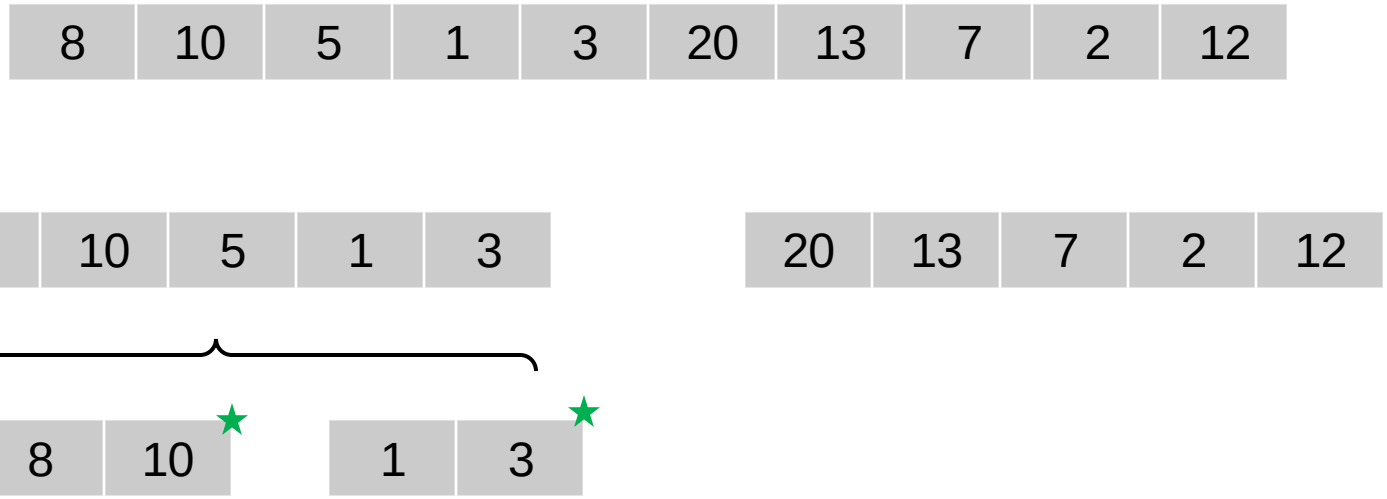
1	3

8	10	5

Merge-sort Example



Merge-sort Example



Merge-sort Example

8	10	5	1	3	20	13	7	2	12

1	3	5	8	10

20	13	7	2	12

20	13	7

2	12

Merge-sort Example

8	10	5	1	3	20	13	7	2	12

1	3	5	8	10 ★

2	7	12	13	20 ★

Merge-sort Example

1	2	3	5	7	8	10	12	13	20 ★

Iterative Merge Sort

- We can remove the recursion using stacks (Straight-forward)
- We can also eliminate the stack using a bottom-up approach

Bottom-up Merge-sort