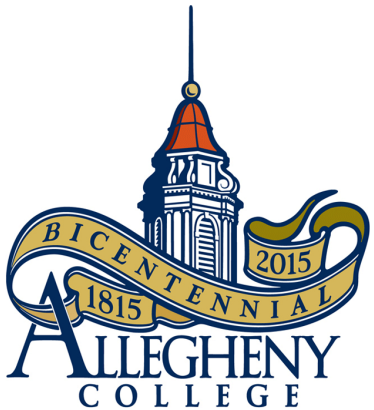
CS202 - Algorithm Analysis

Merge Sort

Aravind Mohan

Allegheny College

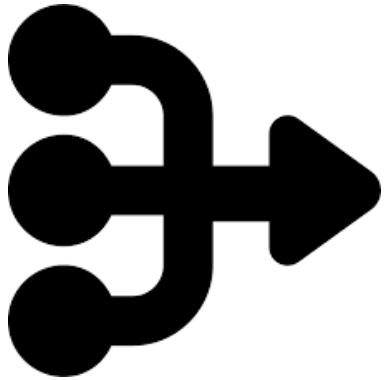
April 5, 2021



Discussion Based On ...

Sedgewick 2.2 Merge Sort

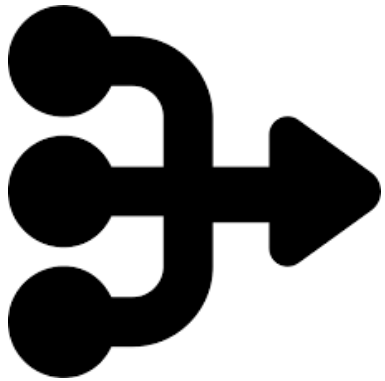
Merge Sort Algorithm



Strategy:

- **Divide:** if S has at least two elements, remove all the elements from S and put them into two sequences S_1 and S_2 , each containing about half of the elements of S.(i.e,. S_1 contains the first floor $(n/2)$ elements and S_2 contains the remaining floor $(n/2)$ elements.
- **Conquer:** Sort sequences S_1 and S_2 using Merge Sort.
- **Combine:** Put back the elements into S by merging the sorted sequences S_1 and S_2 into one sorted sequence.

Merge Sort Algorithm



Characteristics:

- sort out of ”place”, i.e., does require an additional array
- uses divide and conquer principle
- worst case running time is $O(n \times log(n))$

Merge Sort Algorithm

Merge Procedure (linear)

Algorithm - Merge(A, p, m, r)

Input: an n-element un-sorted array A of integer values, a lower bound p of the array A, and a pivot r in the array A.

Output: an n-element sorted array A of integer values.

Merge Sort Algorithm

Merge Procedure (linear)

$n_1 \leftarrow m - p$
$n_2 \leftarrow r - m$
Initialize Two Arrays L of size $n_1 + 1$ and R of size $n_2 + 1$
for i = 0 to n_1 **do**
    L[i] $\leftarrow$ A[p+i]
end for
for j = 0 to n_2 **do**
    R[j] $\leftarrow$ A[m+j]
end for
$L[n_1 + 1] \leftarrow \infty$ and $R[n_2 + 1] \leftarrow \infty$
Initialize $i, j \leftarrow 0$
for k = p to r **do**
    if L[i] $\leq$ R[j] **then**
        A[k] $\leftarrow$ L[i]
        i $\leftarrow$ i+1
    else
        A[k] $\leftarrow$ R[j]
        j $\leftarrow$ j+1
    end if
end for

Merge Sort Algorithm

MergeSort Procedure (logarithmic)

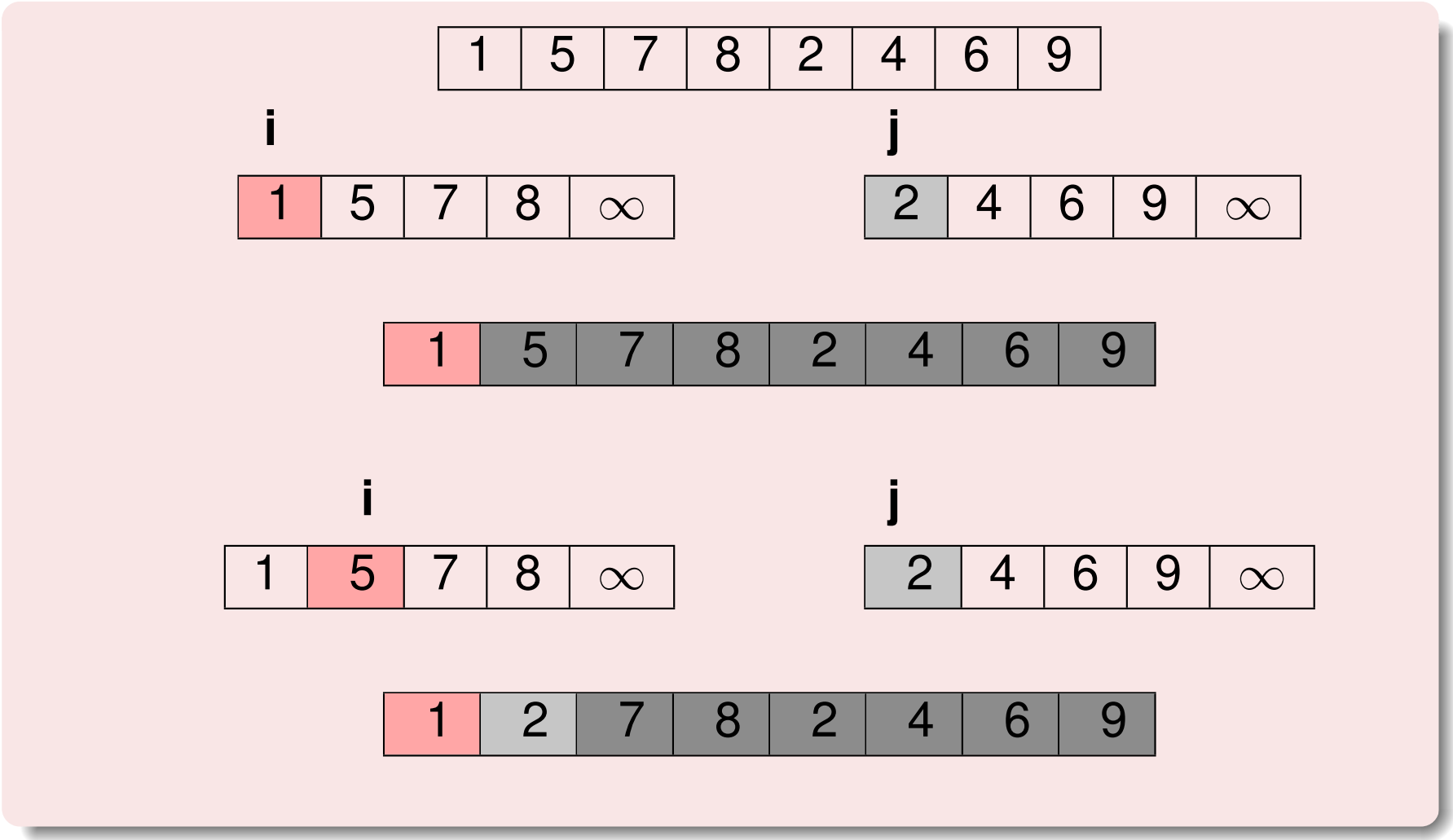
Algorithm - MergeSort(A, p, r)

Input: an n-element un-sorted array A of integer values, a lower bound p of the array A, and a pivot r in the array A.

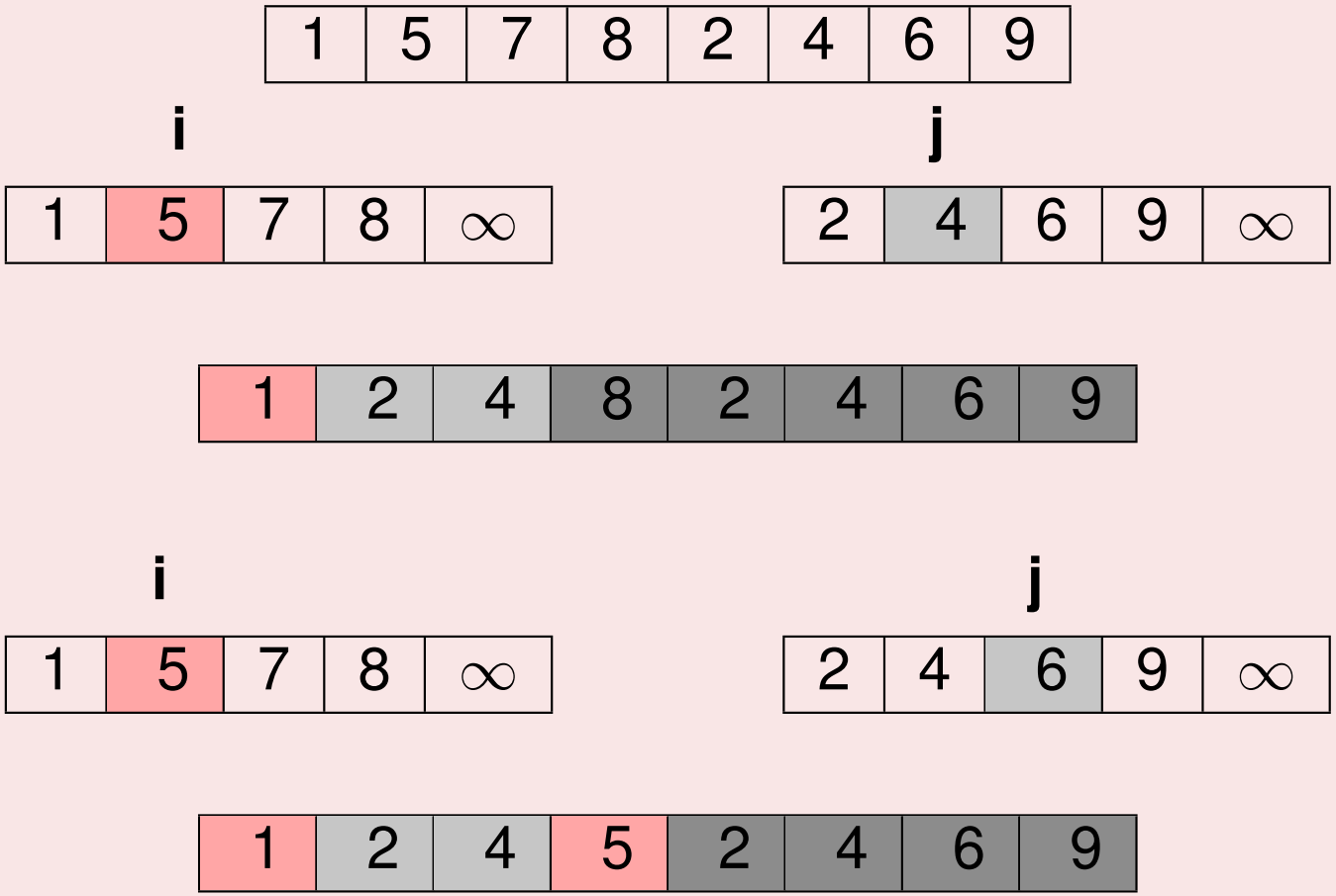
Output: an n-element sorted array A of integer values.

if p $<$ r **then**
  $m \leftarrow$ Floor $(p + r)/2$
  MergeSort(A, p, m)
  MergeSort(A, m+1, r)
  Merge(A, p, m, r)
end if

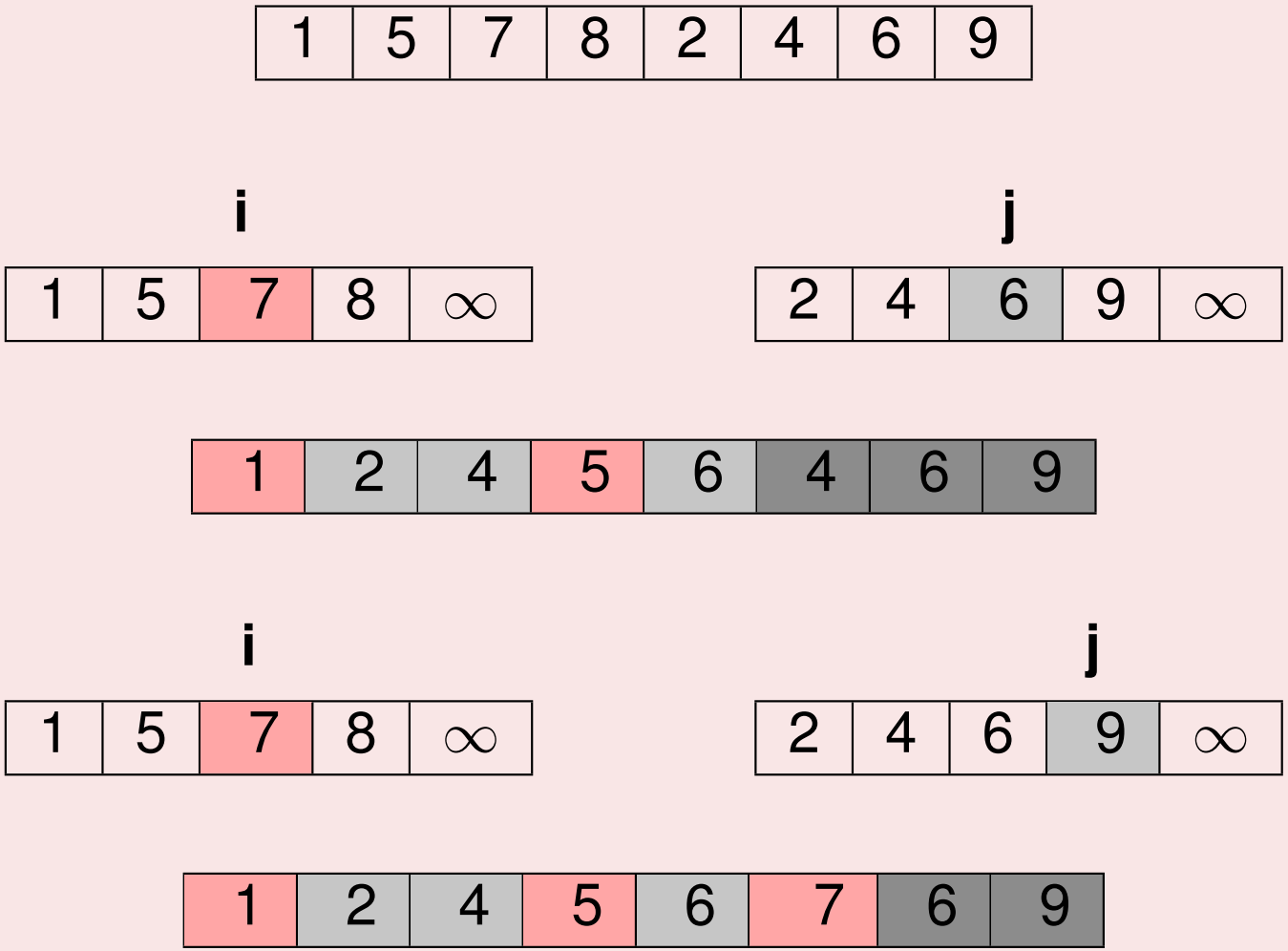
Merge Example



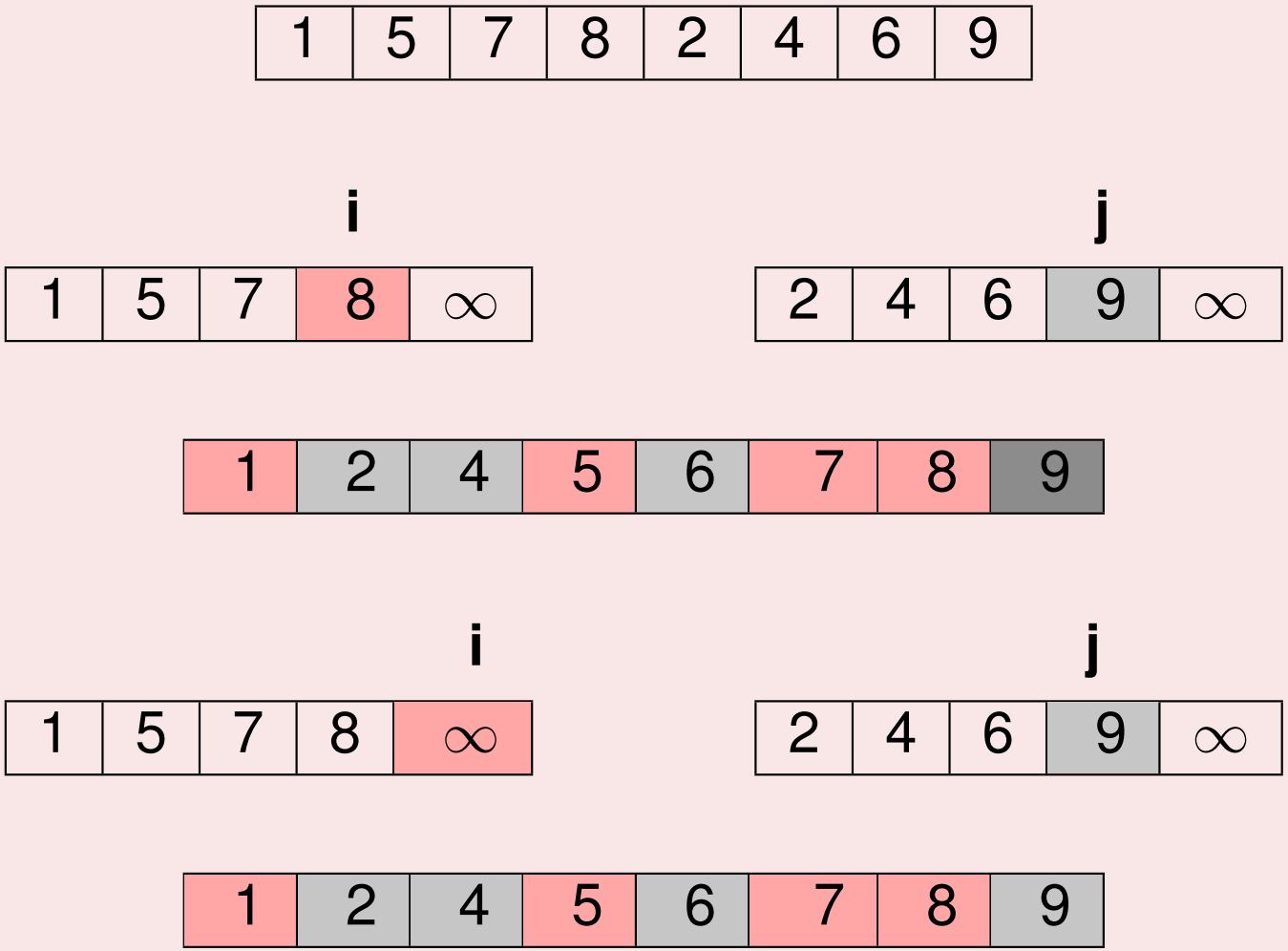
Merge Example



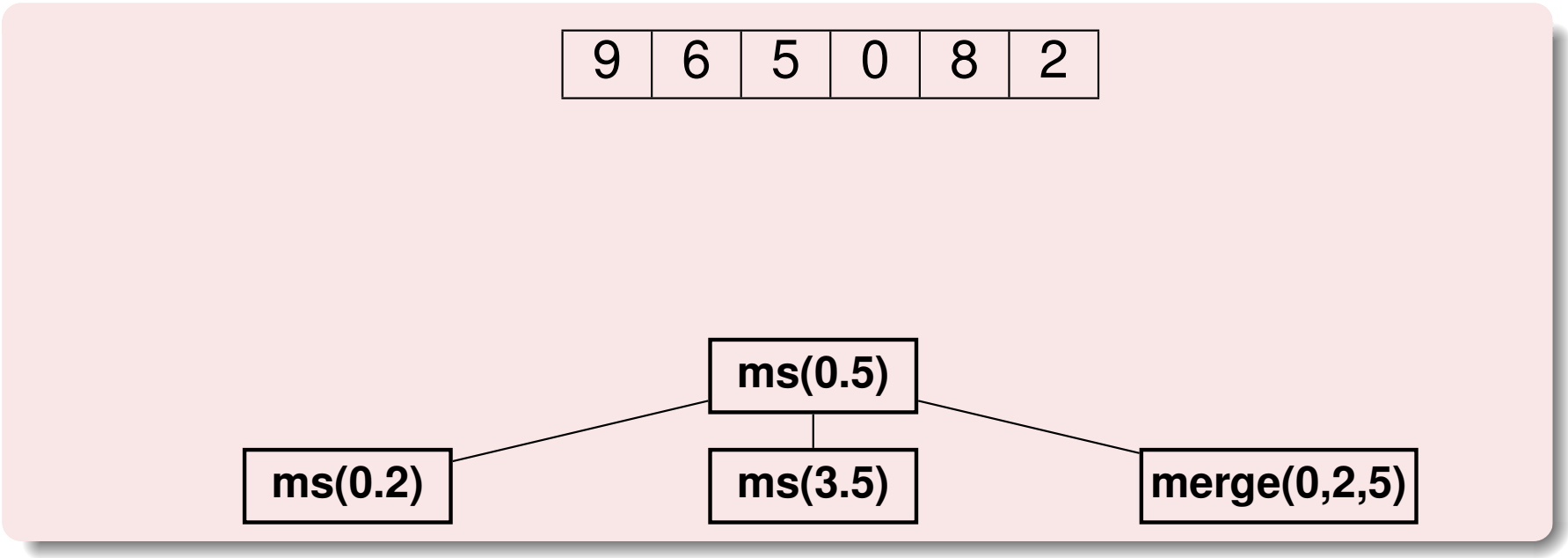
Merge Example



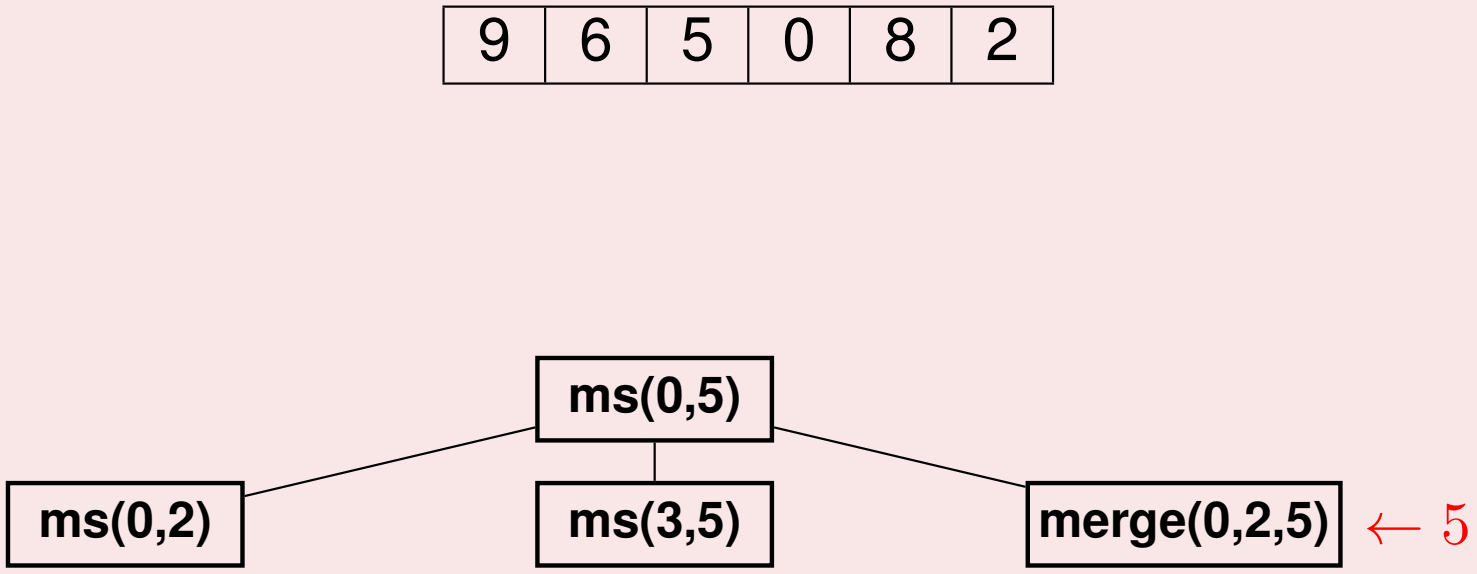
Merge Example



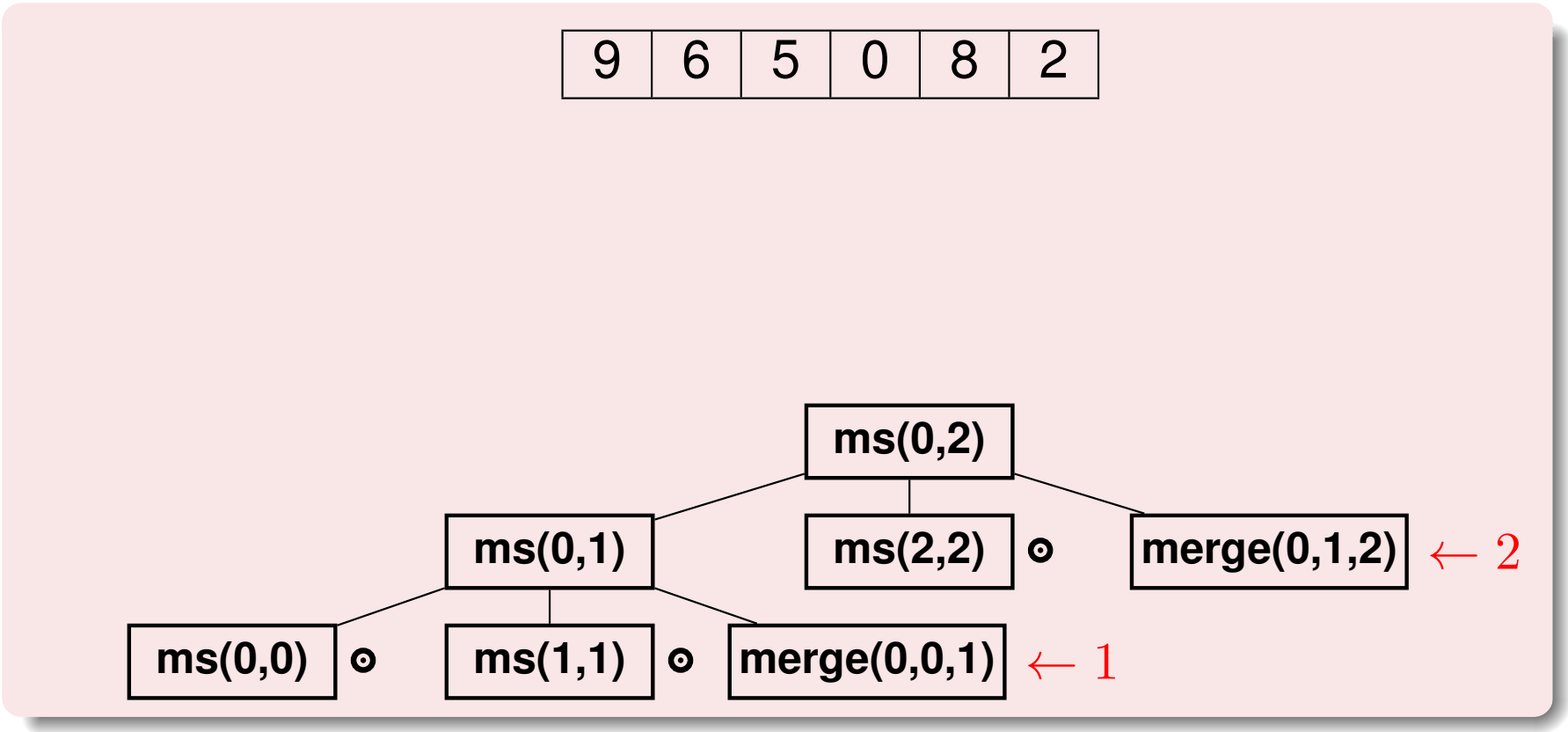
Merge Sort Example



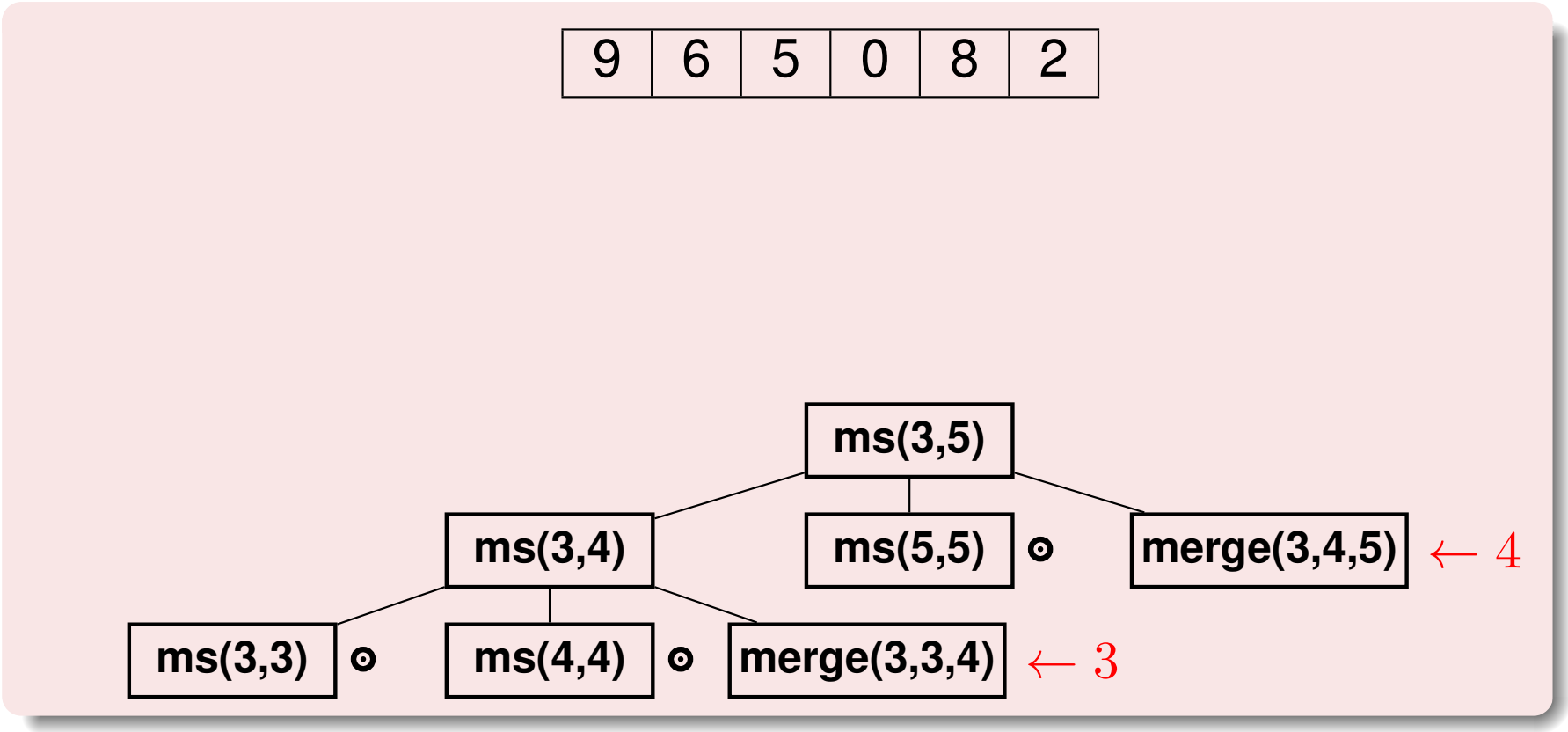
Merge Sort Example



Merge Sort Example



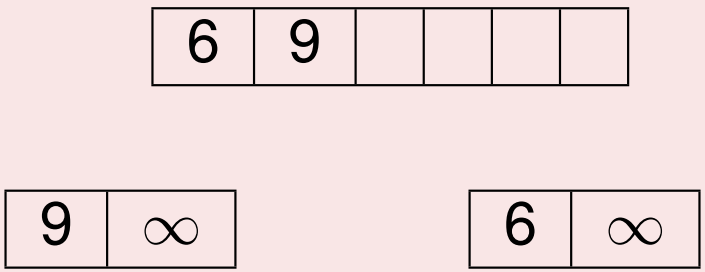
Merge Sort Example



OK done

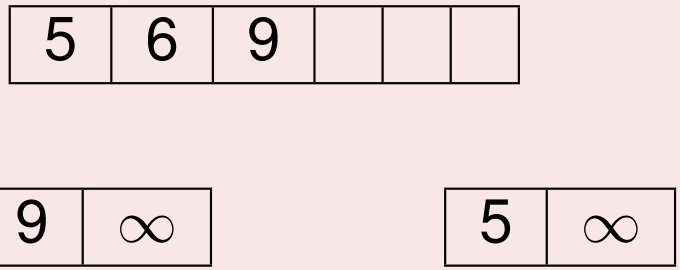
Merge Sort Example

Merge(0,0,1)



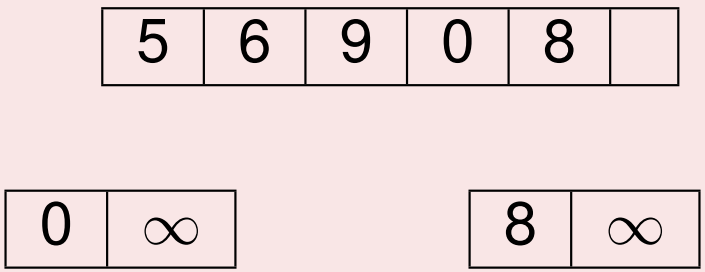
Merge Sort Example

Merge(0,1,2)



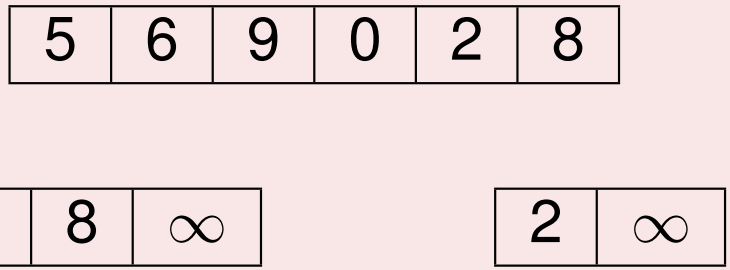
Merge Sort Example

Merge(3,3,4)



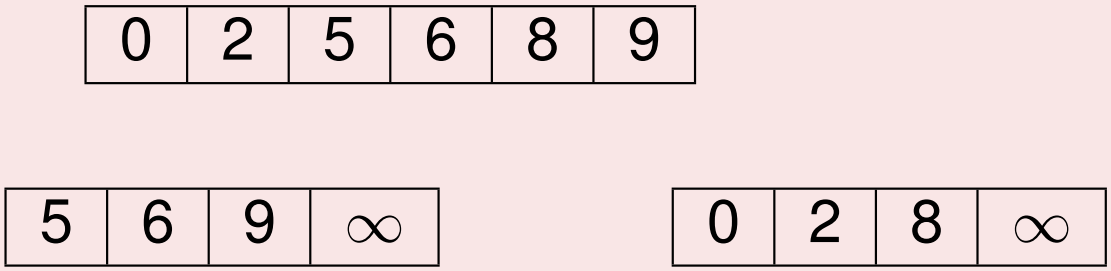
Merge Sort Example

Merge(3,4,5)



Merge Sort Example

Merge(0,2,5)



Done Sorting!

Merge Sort Algorithm - An analysis

- Space complexity: O(n)
- Time complexity: O$(n \times log(n))$ for Worst, Average, and Best case

Next Steps

Tree Data Structure

Reading Assignment

Sedgewick 2.2 Merge Sort

Questions?

Please ask if there are any Questions!