



Design and Analysis of Algorithms

Lecture - 11

Success is always inevitable with Hard Work and Perseverance

N. Ravitha Rajalakshmi

Learning Objective

- Discuss D&C strategy for classical problems
 - Sorting

Sorting

Input: An array A with n elements.

Output: Permutation of Array A where elements are arranged in non-decreasing order.

Initial
Array

130	10	40	8	20	200
-----	----	----	---	----	-----

Sorted
Array

8	10	20	40	130	200
---	----	----	----	-----	-----

Merge Sort

Array
Elements

130	10	40	8	20	200
-----	----	----	---	----	-----

- Split the array into two halves

130	10	40
-----	----	----

8	20	200
---	----	-----

- Sort them independently

10	40	130
----	----	-----

8	20	200
---	----	-----

- Combine

8	10	20	40	130	200
---	----	----	----	-----	-----

Pause & Think

- Can we split the array and do sorting in parallel?

Yes

- Will it affect the final answer ?

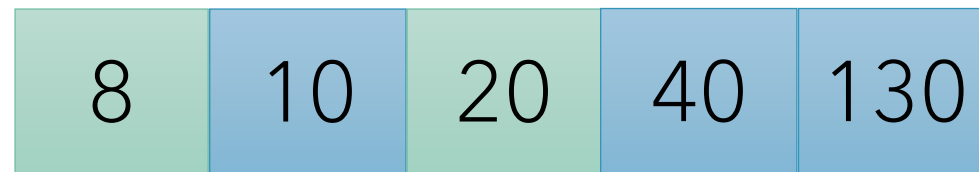
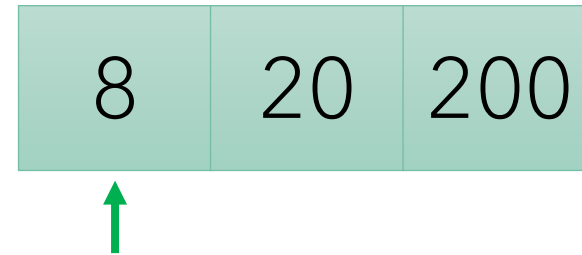
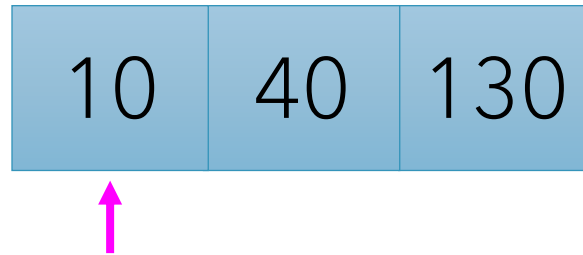
No

- Is there a mechanism for combining the independent solutions?

To be determined

Combine

- Given two sorted arrays, how to combine it such that resultant array is sorted?



Function Merge(A, low, mid, high)

First subarray contains elements from low to mid

Second subarray contains elements from mid + 1 to high

$i \leftarrow \text{low}, k \leftarrow \text{mid} + 1, n \leftarrow \text{low}$ $B[\text{low} \dots \text{high}]$

$\text{while}(i \leq \text{mid} \ \&\& \ k \leq \text{high})\{$

$\text{if}(A[i] \leq A[k])\{$

$B[n] \leftarrow A[i]$

$i++$

$n++$

$\}$

$\text{else}\{$

$B[n] \leftarrow A[k]$

$k++$

$n++$

$\}$

$\}$

Pause & Think

- Is the merge algorithm complete?

No

- After the process, Are all the elements copied to resultant array?

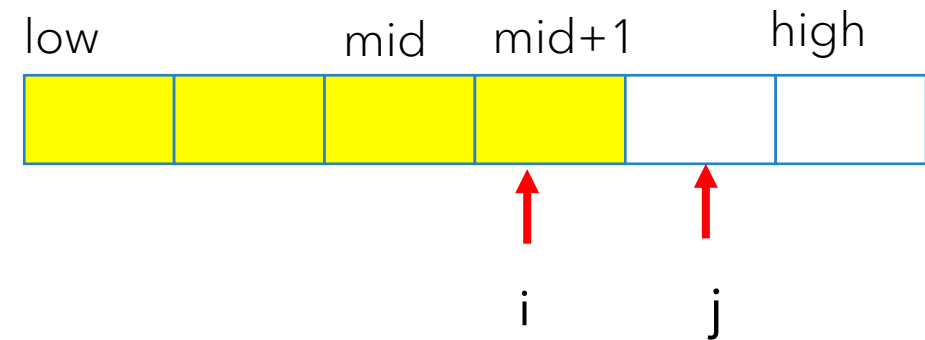
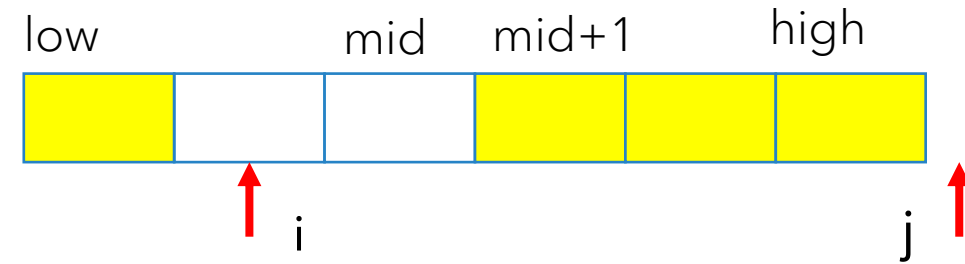
No , 200 is not yet copied to the resultant array


```

if(i<=mid){
    for(int m=i ; m<=mid ; m++){
        B[n] <- A[m]
        n++
    }
}
if(k<=high){
    for(int m=k; m<=high ; m++){
        B[n] <- A[m]
        n++
    }
}

```

Copy the array elements from B to A



The element is copied to resultant array



The element is not copied to the resultant array

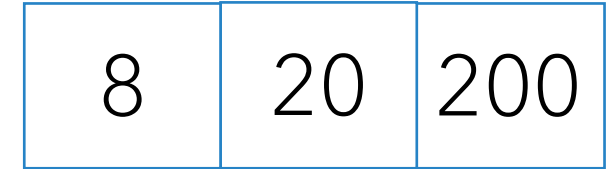
Divide



Divide



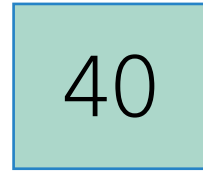
Divide



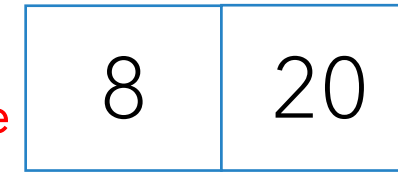
Divide



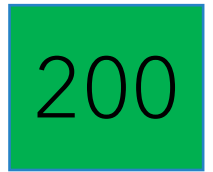
Solve



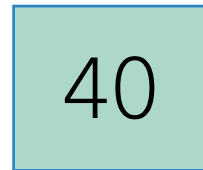
Divide



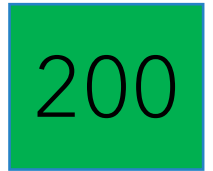
Solve



Solve



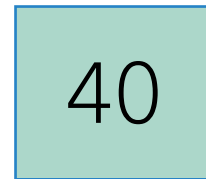
Solve



Combine



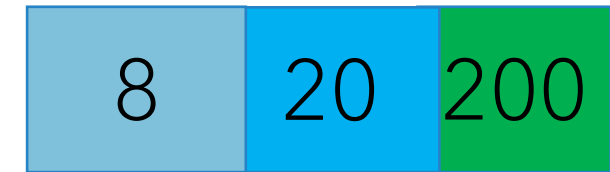
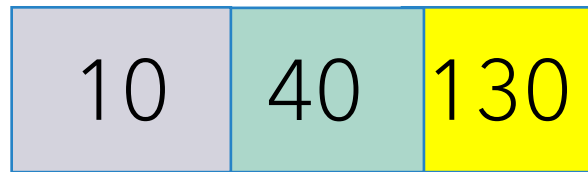
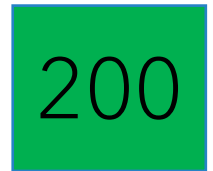
Combine



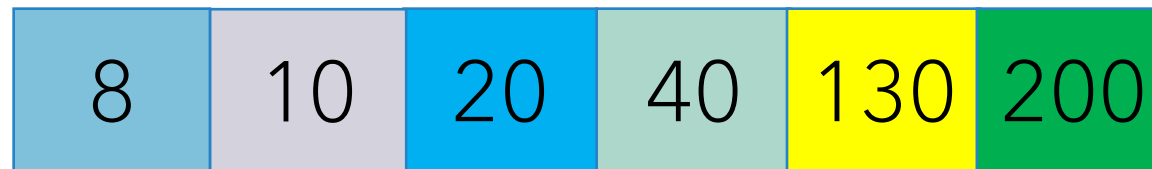
Combine



Combine



Combine



Function MergeSort(A, low, high)

If the array contains more than one element

if (low <= high) {

Divide the array into two subarrays

mid = (low + high)/2

MergeSort(A, low, mid)

MergeSort(A, mid+1, high)

#Combine the solutions

Merge(A, low, mid, high)

}

Analyzing Time Complexity

- Recurrence Relation
- Input size : n
- Basic Operation : Comparison
- Time Complexity : $T(n)$

$$T(1) = 0$$

$$T(n) = 2T\left(\frac{n}{2}\right) + n \text{ (Cost of Merge)}$$

- Using Master's theorem , $T(n) = O(n \log n)$

Pause & Think

- Whether the algorithm is Stable ?

Yes Order of elements with equal values is retained after sorting

- Whether the algorithm is in-place?

No

Extra memory equal to the size of the array is used in merge

Summary

- How D&C is applied for Merge Sort

Thank You
Happy Learning

Success is always inevitable with Hard Work and Perseverance