

Algorithm Design & Problem Solving: **Mergesort**



Contents



1

Merge sort

2

Merge sort algorithm

Merge Sort



❖ Use the merge sort to sort these numbers

4	2	6	8	1	3	7	5
---	---	---	---	---	---	---	---

Merge Sort (Divide & Conquer)



4	2	6	8	1	3	7	5
---	---	---	---	---	---	---	---

Merge Sort (Divide & Conquer)



4 2 6 8 1 3 7 5

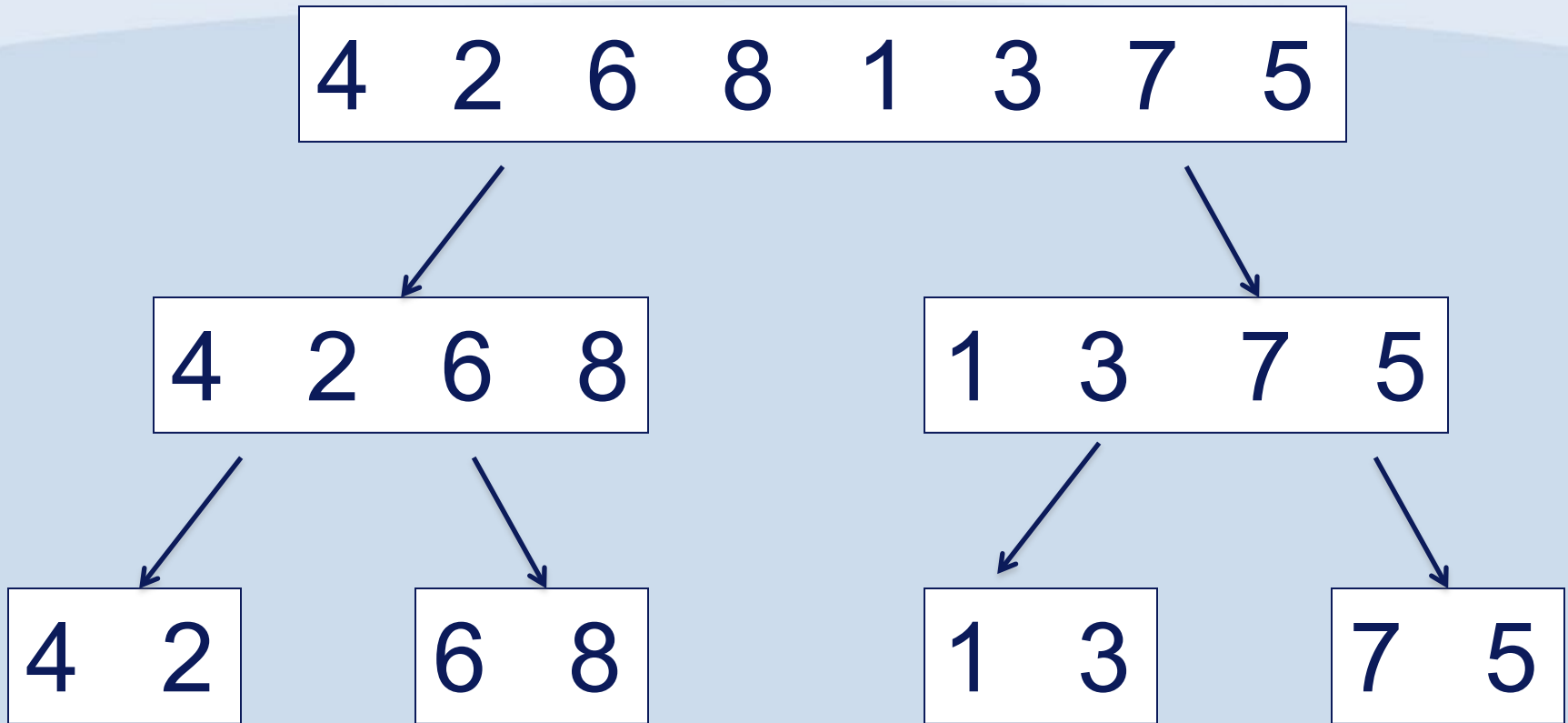


4 2 6 8

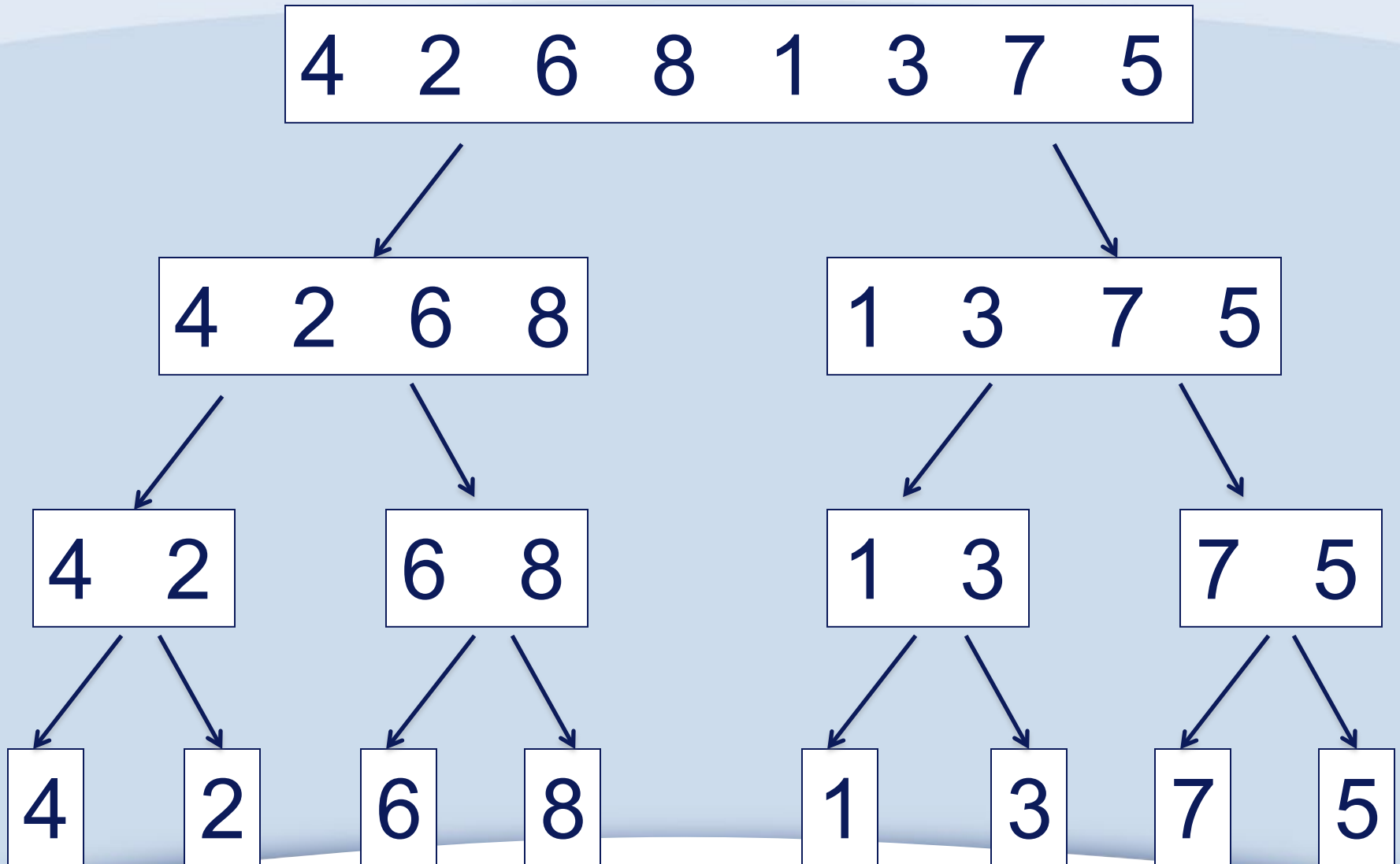


1 3 7 5

Merge Sort (Divide & Conquer)



Merge Sort (Divide & Conquer)



Merge Sort (Combine)



4

2

6

8

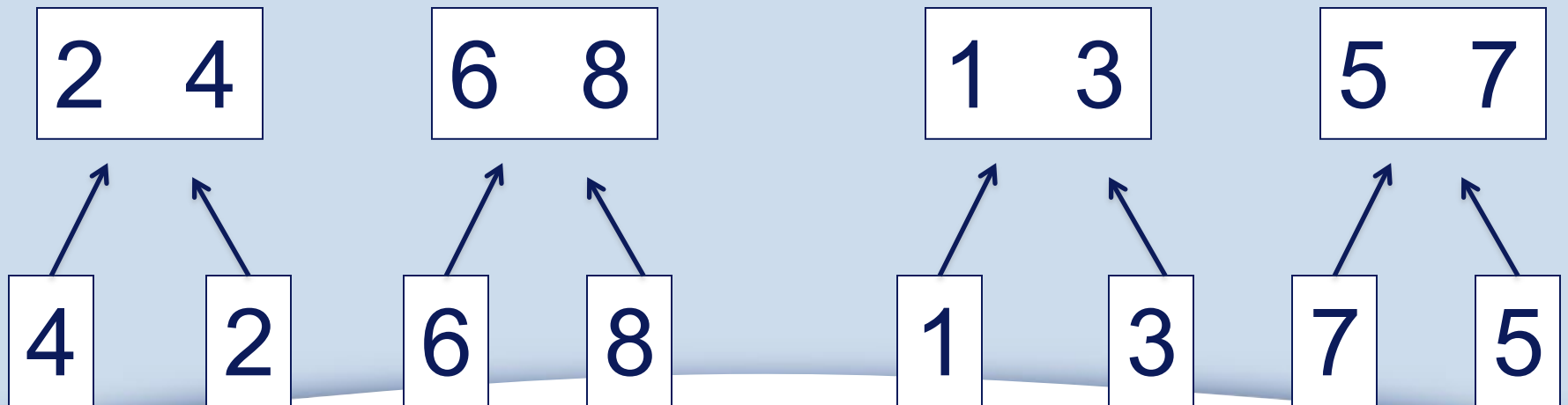
1

3

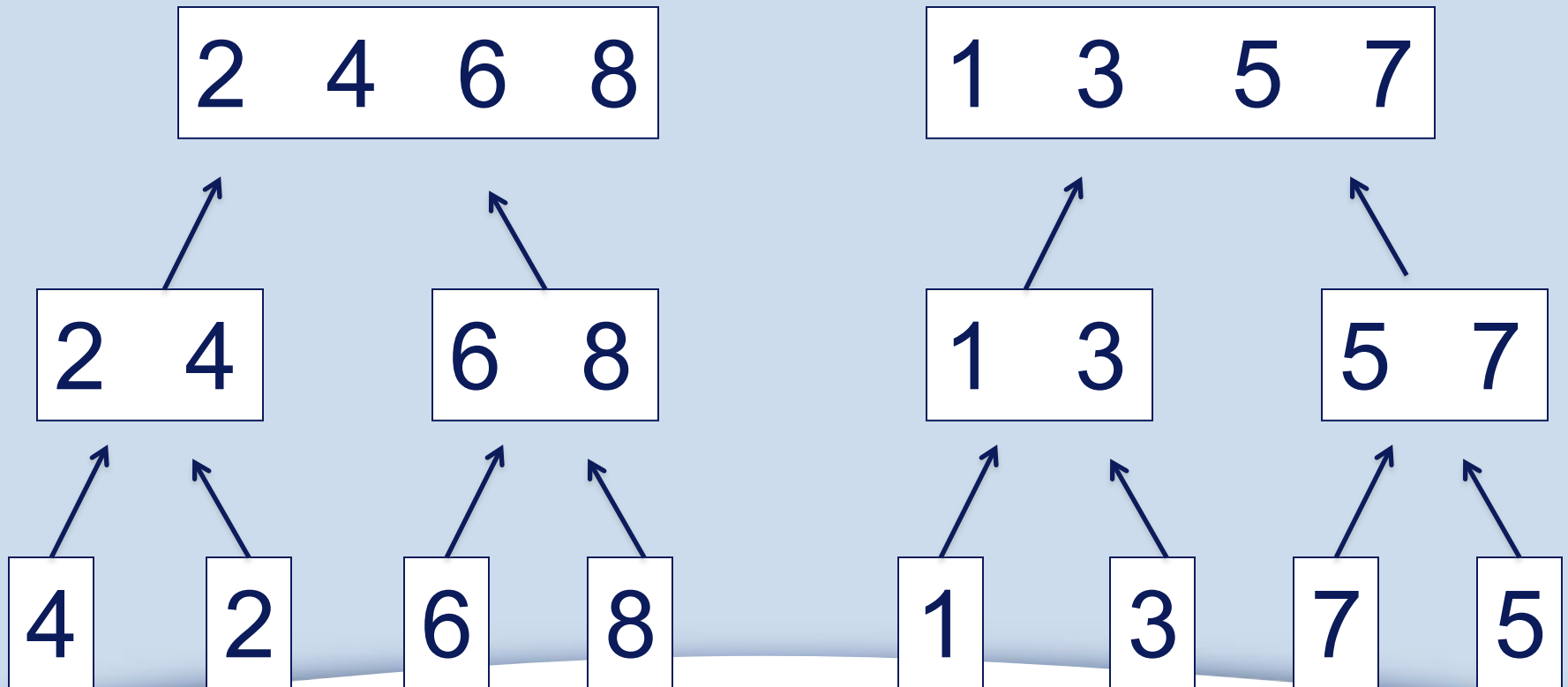
7

5

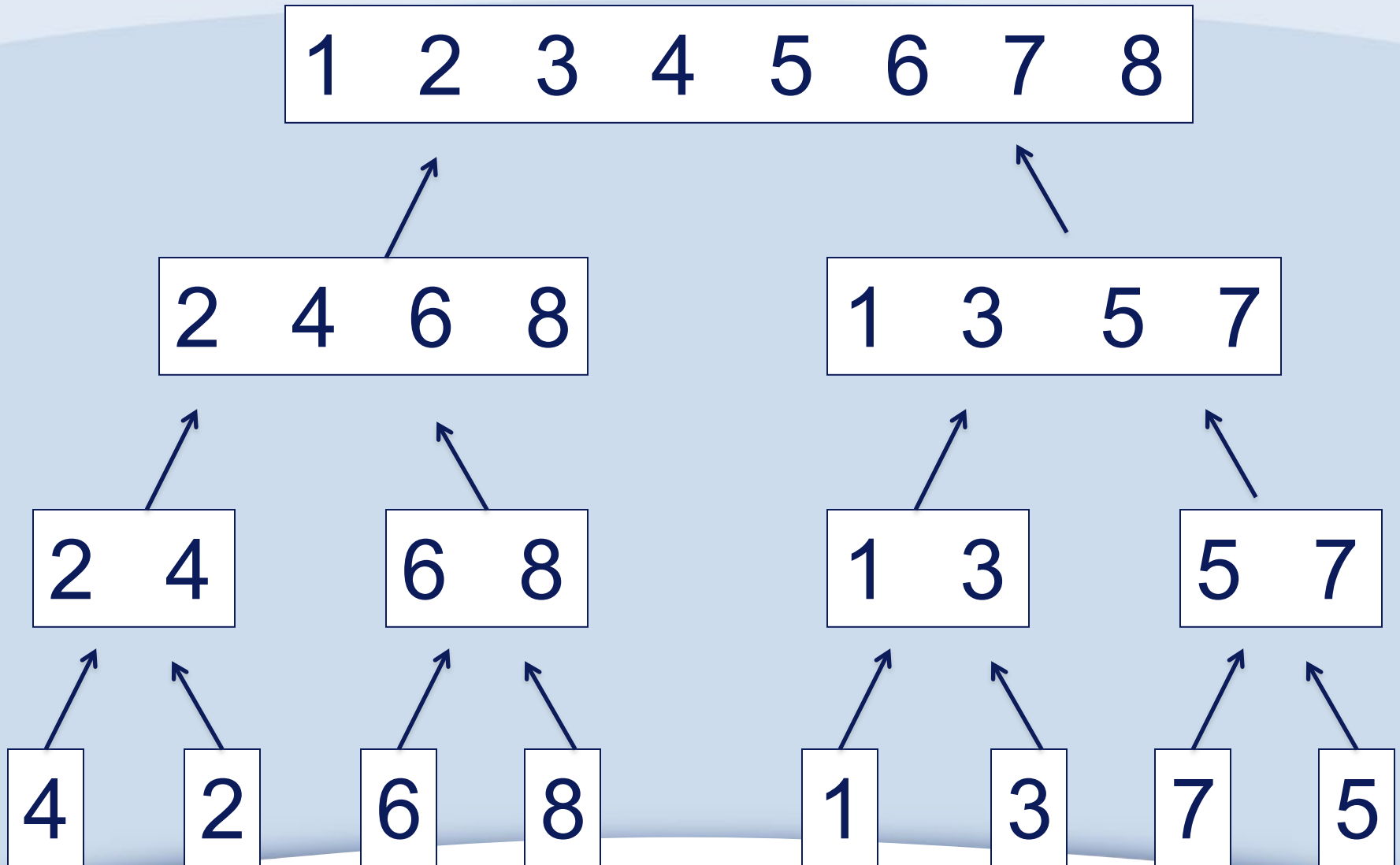
Merge Sort (Combine)



Merge Sort (Combine)



Merge Sort (Combine)



Merge Sort: An algorithm



How does merge sort work ... in English?

Reminder: What is Recursion?



When one function calls ITSELF directly or indirectly.

Reminder: What is Recursion?



- ❖ New mode of thinking.
- ❖ Powerful programming tool.
- ❖ Divide-and-conquer paradigm.

Reminder: Recursive Factorial



Factorial (n)

if $n=1$ or $n=0$

return 1

else

return $n * \text{Factorial}(n-1)$

Merge Sort: An algorithm



On input of n elements:

If $n < 2$

Return

Else

Sort left half of elements.

Sort right half of elements.

Merge sorted halves.

Merge Sort

1. Divide

Finds middle of array

2. Conquer

3. Combine

Merge Sort

1. Divide

2. Conquer

Recursively solve two
sub problems

3. Combine

Merge Sort

1. Divide

2. Conquer

3. Combine

Merge to recreate array

Merge Sort & Big O



❖ Merge Sort (Divide & Conquer) $O(\log n)$

❖ Merge (Combine)

$O(n)$



$O(n \log n)$

Merge Sort: The algorithm ..in more detail



Input: An array **A** storing items from index **low** to **high**

Output: **A** sorted in ascending order



1. Merge_Sort (A, low, high):

2. If $n < 2$

3. Return

4. Else

5. mid = (low+high)/2

Divide

6. Merge_Sort(A, low, mid)

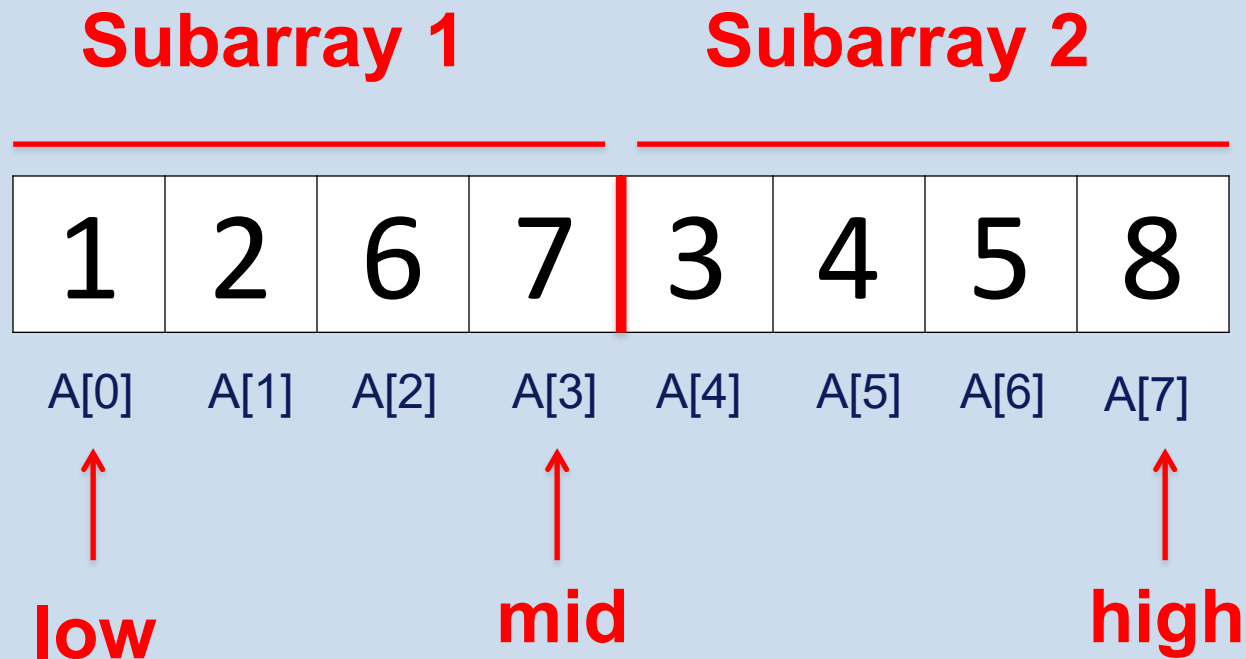
Conquer

7. Merge_Sort(A, mid+1, high)

8. Merge(A, low, mid, high)

Combine

Combining/Merging: Merge Sort



Part II: Merge Sort



Merge the subarrays until the end of one is reached.

1. Compare first element of both subarrays. Place the smallest of the 2 first elements into a temp array and move to next space in subarray

Array A

1	2	6	7		3	4	5	8
---	---	---	---	--	---	---	---	---

Array tempArray

1

Array A

1	2	6	7		3	4	5	8
---	---	---	---	--	---	---	---	---

Part II: Merge Sort



A

1	2	5	6		3	4	7	8
---	---	---	---	--	---	---	---	---

tempArray

1

A

1	2	5	6		3	4	7	8
---	---	---	---	--	---	---	---	---

tempArray

1	2
---	---

A

1	2	5	6		3	4	7	8
---	---	---	---	--	---	---	---	---

tempArray

1	2	3
---	---	---

Part II: Merge Sort



1. Merge the subarrays until the end of one is reached.
 1. Compare first element of both subarrays. Place the smallest of the 2 first elements into a temp array and move to next space in subarray
2. If end of left subarray reached
 - Copy the right subarray into tempArray
 - Otherwise, copy the left subarray into tempArray

1	2	5	6		3	4	7	8
---	---	---	---	--	---	---	---	---

Part II: Merge Sort



A

1	2	5	6		3	4	7	8
---	---	---	---	--	---	---	---	---

tempArray

1	2	3	4	5	6
---	---	---	---	---	---

A

1	2	5	6		3	4	7	8
---	---	---	---	--	---	---	---	---

tempArray

1	2	5	6	3	4	7	8
---	---	---	---	---	---	---	---

Part II: Merge Sort



1. Merge the subarrays until the end of one is reached.
 1. Compare first element of both subarrays. Place the smallest of the 2 first elements into a temp array and move to next space in subarray
2. If end of left subarray reached
 - Copy the left subarray
 - Otherwise, copy the right subarray
3. Copy values back into original array

Part II: Merge Sort



tempArray

1	2	5	6	3	4	7	8
---	---	---	---	---	---	---	---



A

1	2	5	6	3	4	7	8
---	---	---	---	---	---	---	---



**Write the algorithm for the Merge(A,
low, mid, high) function**

Merge Sort

MERGE SORT

Best

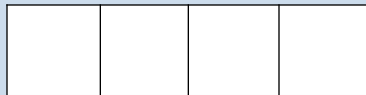
$\Omega(n \log n)$

Average

$\Theta(n \log n)$

Worst

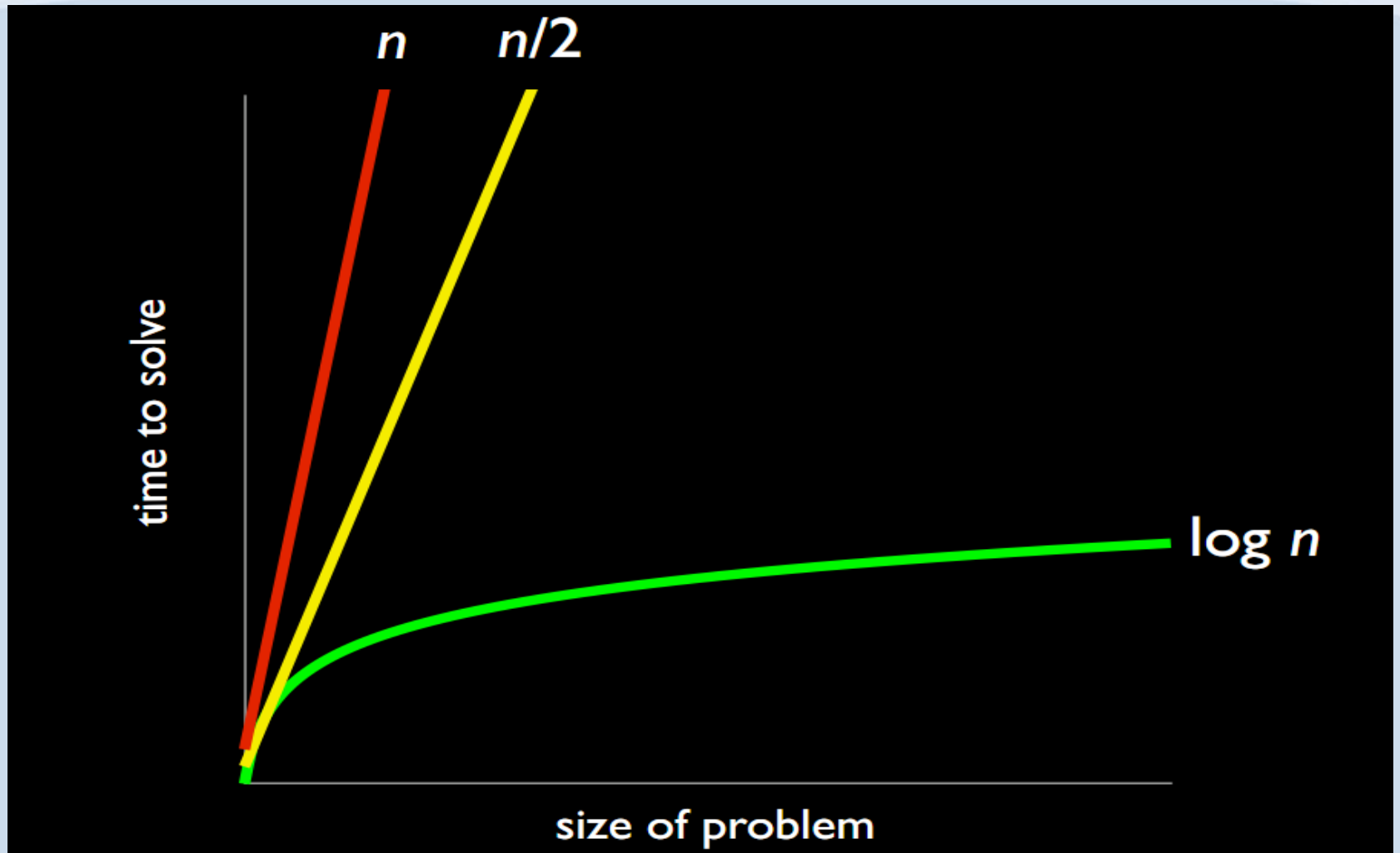
$O(n \log n)$



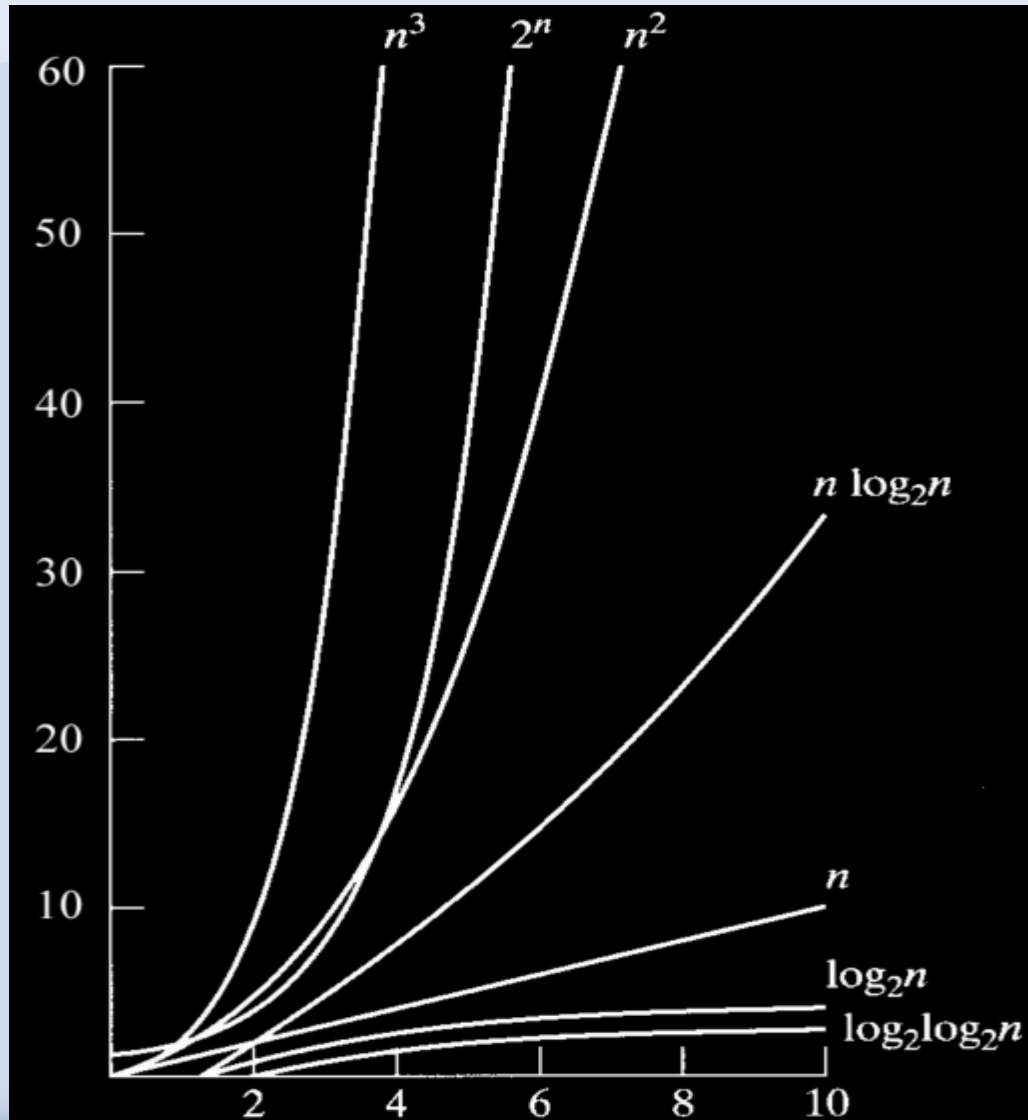
Array

Recursion Divide & Conquer

Remember this graph ...



Graph of 3 types of searches



Merge Sort



❖ Sort the following numbers using Merge sort

3	41	52	26	38	57	9	49
---	----	----	----	----	----	---	----

Thank You !

