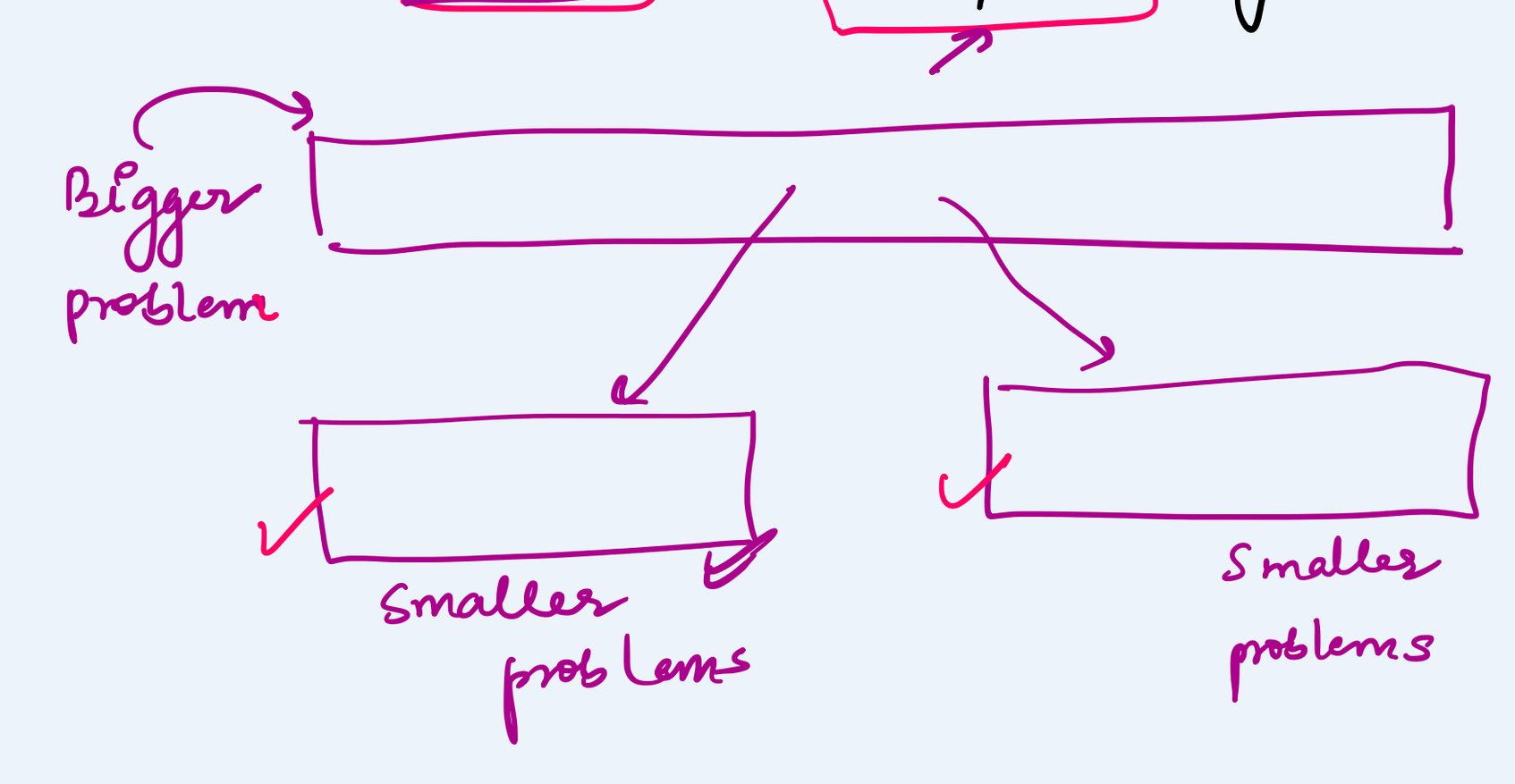
Wednesday, 3 March 2021 4:48 PM

Divide & Conquer / Combining algorithm



Ques: Given an array of n integers, find the max element of the array.

3	2	6	7	1	3	2	-1	10

1 person

Approach ①

→ Iterate over each and every element and keep track of the max ele so far. [linear search]

3	2	6	7	1

ans 1 = 7

3	2	-1	10

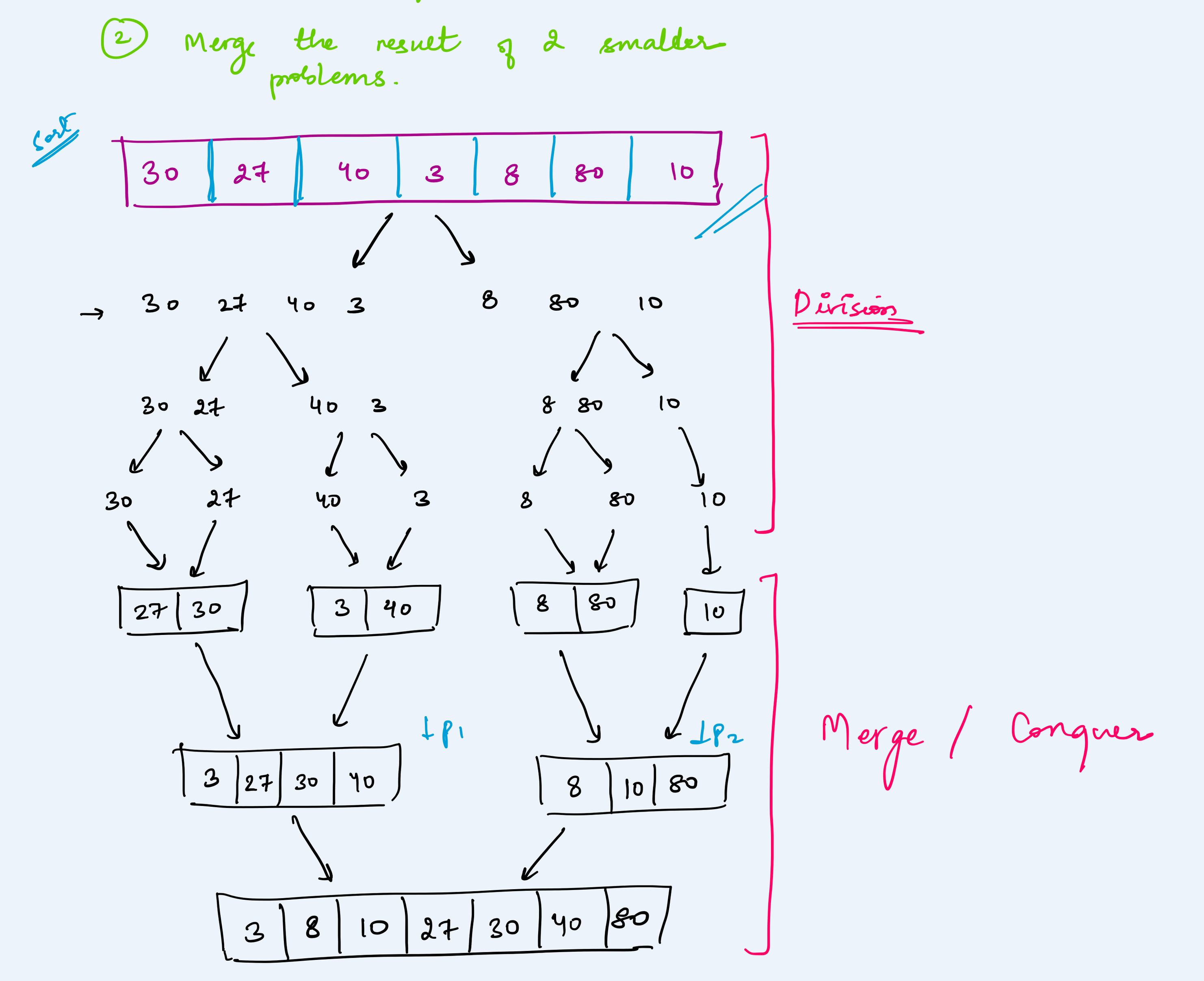
ans2 = 10

```
if (ans1 > ans2) {
    res = ans1;
} else {
    res = ans2;
}
```

Merge sort

① Divide the problem into 2 halves

② Merge the result of 2 smaller problems.



Divide & Conquer

→ Binary search
→ Ternary search
→ Quick sort
→ Merge sort

① Find the middle part

$m = l + (r - l)/2$

② Call merge sort for first halves

ms(arr, l, m) → range

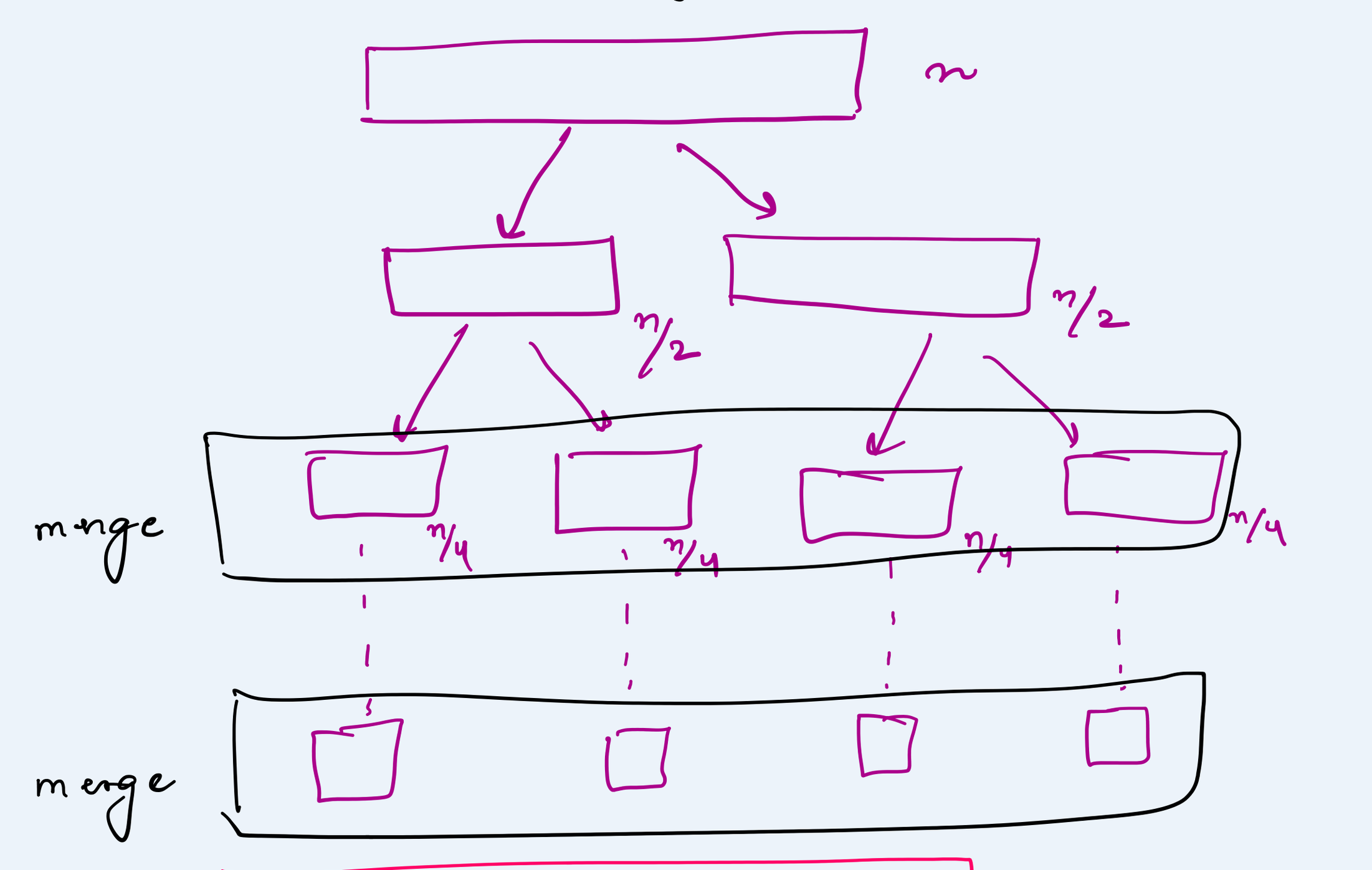
③ Call merge sort for second half

ms(arr, m+1, r) → range

④ Merge the two halves that were sorted in step 2 and step 3.

merge(arr, l, m, r) → entire range

Time Complexity



$T(n) = 2T\left(\frac{n}{2}\right) + O(n)$ → ①

Recurrence Relation

→ Substitution method

$T(n) = 2T\left(\frac{n}{2^2}\right) + \frac{n}{2}$

$T(n) = 2\left[2T\left(\frac{n}{2^2}\right) + \frac{n}{2}\right] + n$

$T(n) = 2^2 + T\left(\frac{n}{2^2}\right) + n + n$ — ②

$T\left(\frac{n}{2^2}\right) = 2T\left(\frac{n}{2^3}\right) + \frac{n}{2^2}$

$T(n) = 2^2\left[2T\left(\frac{n}{2^3}\right) + \frac{n}{2^2}\right] + 2n$

$T(n) = 2^3 T\left(\frac{n}{2^3}\right) + 3n$

$T(n) = 2^k T\left(\frac{n}{2^k}\right) + kn$

$T\left(\frac{n}{2^k}\right) = T(1)$

$\therefore \frac{n}{2^k} = 1 \Rightarrow n = 2^k$

$k = \log n$

$T(n) = 2^k T(1) + kn$

$k = \log n$

$T(n) = n \times 1 + \log n$

$T(n) = n\log n$

Merge Sort

TC → Worst case → $O(n\log n)$
Best case → $O(n\log n)$
Avg case → $O(n\log n)$

SC → $O(n)$

Q → Inversion Count

Inversion count for an array tells how far the array is from being sorted.

Formally speaking, two elements a[i] and a[j] form an inversion if a[i] > a[j] and (i < j)

Eg: arr [8 4 2 1]

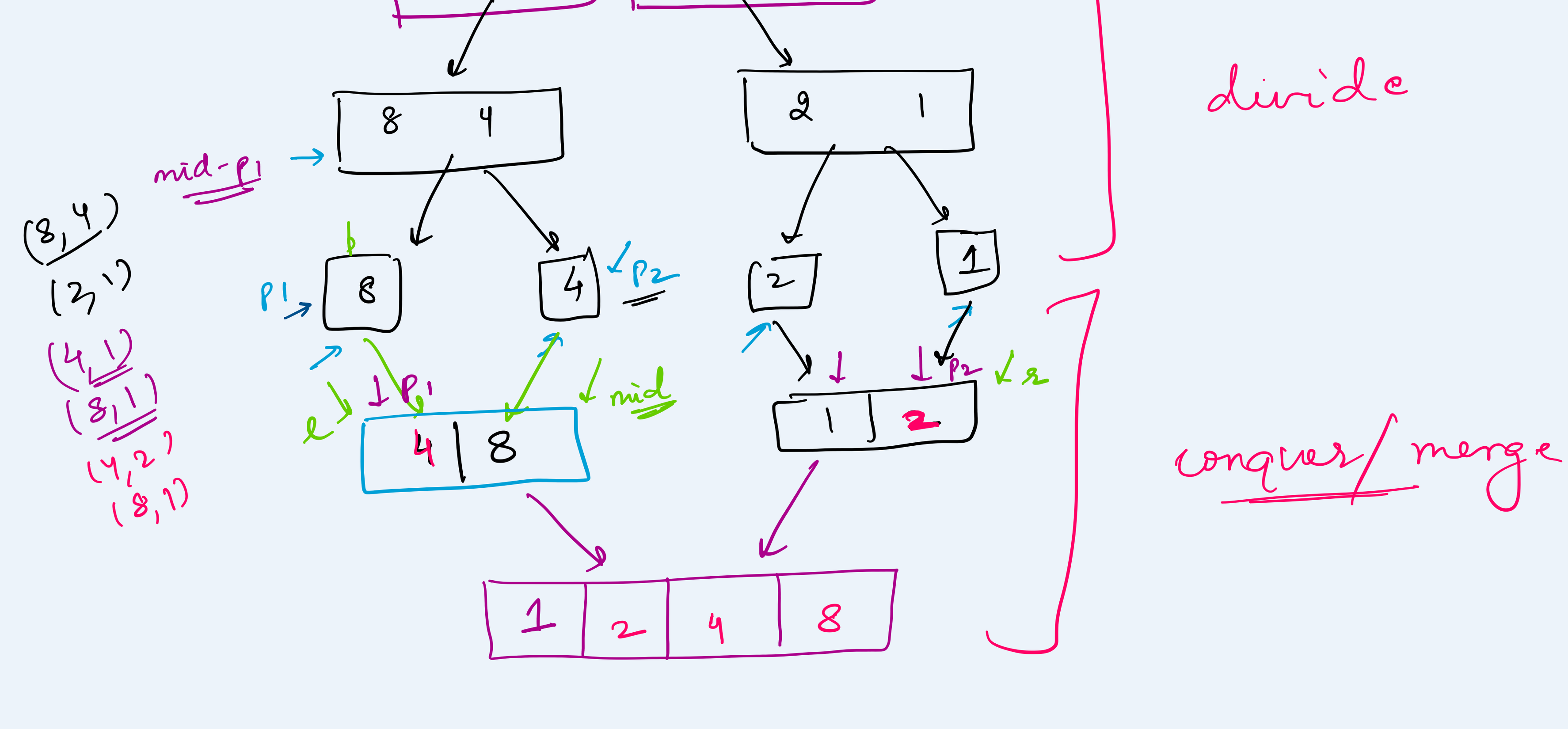
output → 6

Explanation: (8, 4), (8, 2), (4, 2), (8, 1), (4, 1), (2, 1)

Eg: arr[] = {3, 1, 2}

output → 2

Explanation: (3, 1), (3, 2)



① Inversion count of left subproblem + Inversion count of right subproblem + Inversion count during merge.