

Q Given an integer array. Find the count of inversion pairs in the array.

Inversion Pair: (i, j) s.t.

- 1) $i < j$
- 2) $A[i] > A[j]$

Eg $A = [8, 3, 4]$ $(0, 1)$, $(0, 2)$
 ~~$(1, 2)$~~

Ans = 2

$A = [4, 5, 1, 2, 6, 3]$

$(0, 2)$ $(1, 2)$ $(4, 5)$
 $(0, 3)$ $(1, 3)$
 $(0, 5)$ $(1, 5)$

Ans = 7

$A = [4, 4, 4, 4, 4]$

Ans = 0

Solⁿ \rightarrow Brute force

\forall pairs (i, j) where $i < j$ check & \uparrow count

```

count = 0;
for (i = 0; i < N; i++) {
    for (j = i+1; j < N; j++) {
        if (A[i] > A[j]) {
            count++;
        }
    }
}

```

$$T.C. = O(N^2)$$

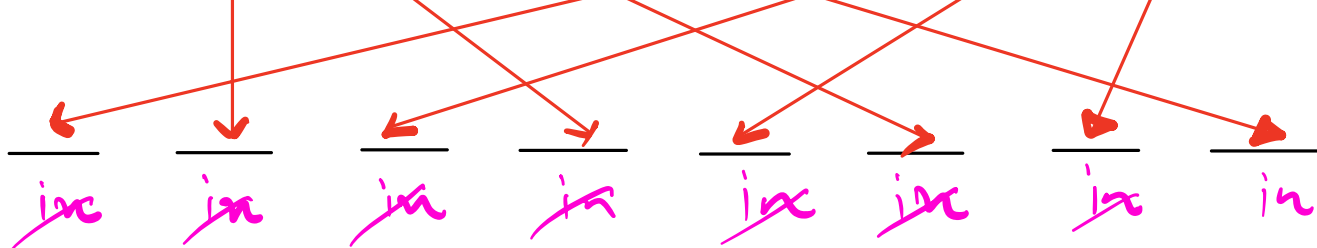
$$S.C. = \underline{\underline{O(1)}}$$

$A = [\overset{0}{\textcircled{4}}, \overset{1}{\textcircled{5}}, \overset{2}{1}, \overset{3}{\textcircled{2}}, \overset{4}{6}, \overset{5}{3}] \Rightarrow$

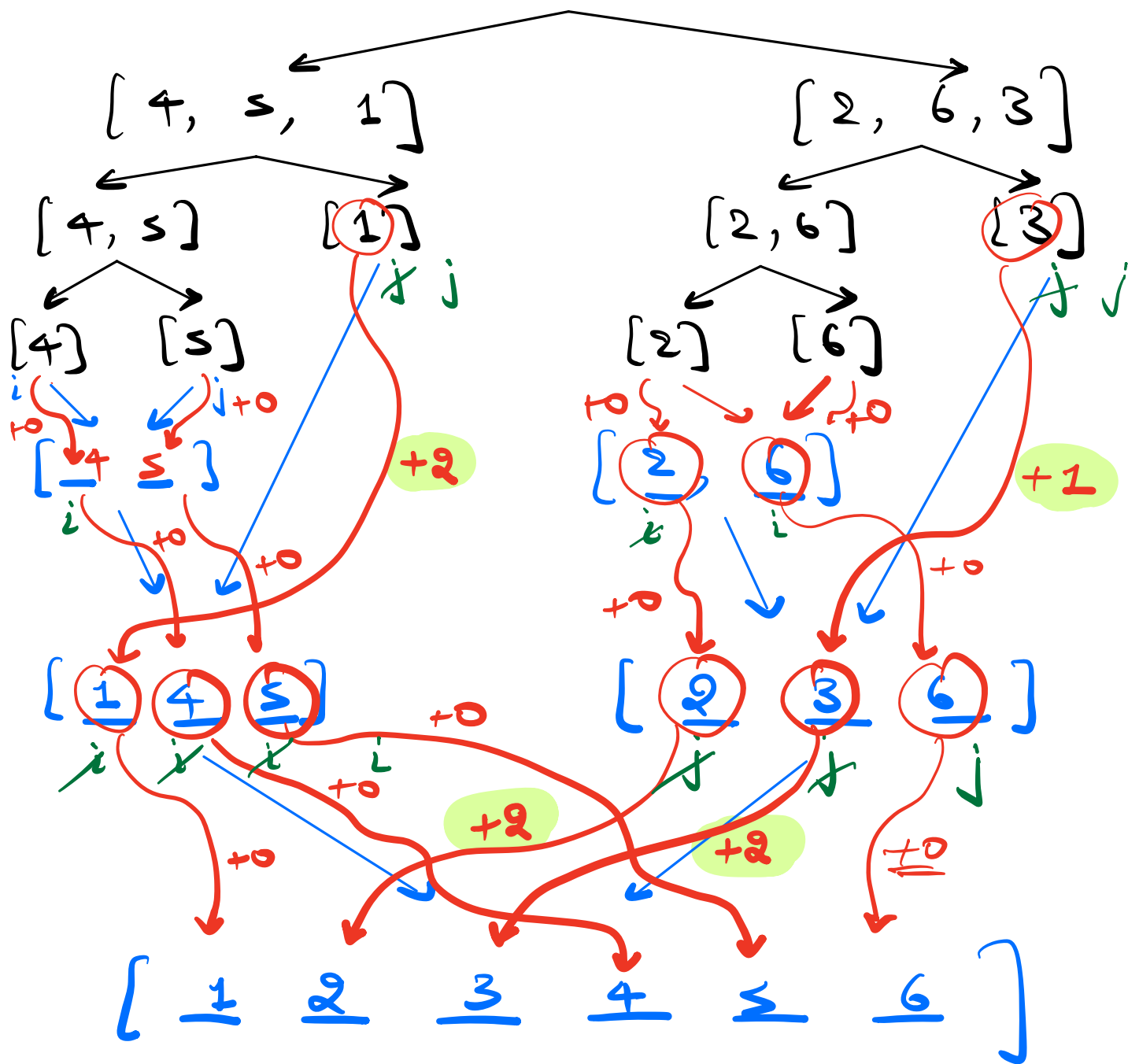
Sorted: $[\underline{1}, \underline{2}, \underline{3}, 4, 5, 6]$

Merge Sort

$[\overset{+4}{\cancel{2}}, \overset{+3}{\cancel{4}}, \overset{+2}{\cancel{7}}, \overset{+1}{\textcircled{9}}]$
 $[\textcircled{1}, \cancel{3}, \cancel{6}, \cancel{8}]$



$$A = [4, 5, 1, 2, 6, 3]$$



$$\text{Ans} = 7$$

$$T.C. = O(N \log N)$$

$$S.C. = O(N)^2$$

H.W.

Code ??

22nd \Rightarrow Friday

25
27
29
1

} off

3
5
6

}

1) Pivot partition

2) Quick Sort

3) Comparison problems

\hookrightarrow B closest points to origin

\hookrightarrow Longest Number

Q Given an array of 0's & 1's in random order.

Segregate all 0's on left & 1's on right

$A = \{ \overset{0}{0}, \overset{1}{1}, \overset{2}{0}, \overset{3}{0}, \overset{4}{1}, \overset{5}{1}, \overset{6}{0}, \overset{7}{1}, \overset{8}{0} \}$

O/P : $\{ \overset{0}{0}, \overset{1}{0}, \overset{2}{0}, \overset{3}{0}, \overset{4}{0}, \overset{5}{1}, \overset{6}{1}, \overset{7}{1}, \overset{8}{1} \}$

Solⁿ

1) Use any sorting algo.

- | | |
|---|------------|
| 1) Bubble Sort | } $O(N^2)$ |
| 2) Selection " | |
| 3) Insertion " | |
| 4) Merge Sort $\Rightarrow O(N \log N)$ | |

Expected T.C. = $O(N)$

\Rightarrow Count Sort $\Rightarrow O(N)$

Idea w/o sorting

First Non 0 found.

\rightarrow

$A = \{ \overset{0}{0}, \overset{1}{0}, \overset{2}{0}, \overset{3}{0}, \overset{4}{0}, \overset{5}{1}, \overset{6}{1}, \overset{7}{1}, \overset{8}{1} \}$

$\text{Next } j$

$\leftarrow \text{0's} \quad \text{1's} \quad \text{2's} \rightarrow$

$T.C. = O(N)$

Code

$i = 0$
 $j = 0$

$\text{while } (A[i] == 0 \text{ \& } i < N) \{$
 $\quad i++;$

$\}$

$\text{for } (j = i+1; j < N; j++) \{$

$\text{if } (A[j] == 0) \{$

$\text{temp} = A[i];$

$A[i] = A[j];$

$A[j] = \text{temp};$

$i++;$

Given an integer array & a pivot
 We need to rearrange the array

Elements smaller than pivot	P	Greater than pivot
-----------------------------------	---	--------------------------

Pivot
 $A = [54, 26, 93, 17, 77, 31, 44, 55, 20]$

$A = [31, 26, 20, 17, 44, 54, 77, 55, 93]$

Sorted $A = [17, 20, 26, 31, 44, 54, 55, 77, 93]$

$[0, i-1] < \text{pivot}$

$A = [26, 17, 31, 44, 20, 93, 54, 55, 77]$

pivot = 54

54

$[i+1, N-1] > p$

H.W. Code ?? \Rightarrow int partition(A, p);

Quick Sort \Rightarrow Divide & Conquer

pivot

A = [54, 26, 93, 17, 77, 31, 44, 55, 20]

Part. A =

0	1	2	3	4	5	6	7	8
31, 26, 20, 17, 44,					54,	77, 55, 93		

Sorted A =

0	1	2	3	4	5	6	7	8
17, 20, 26, 31, 44,					54,	55, 77, 93		

1:1 mapping
 \Downarrow recursive

1:1 mapping
 \Downarrow recursive

Steps:

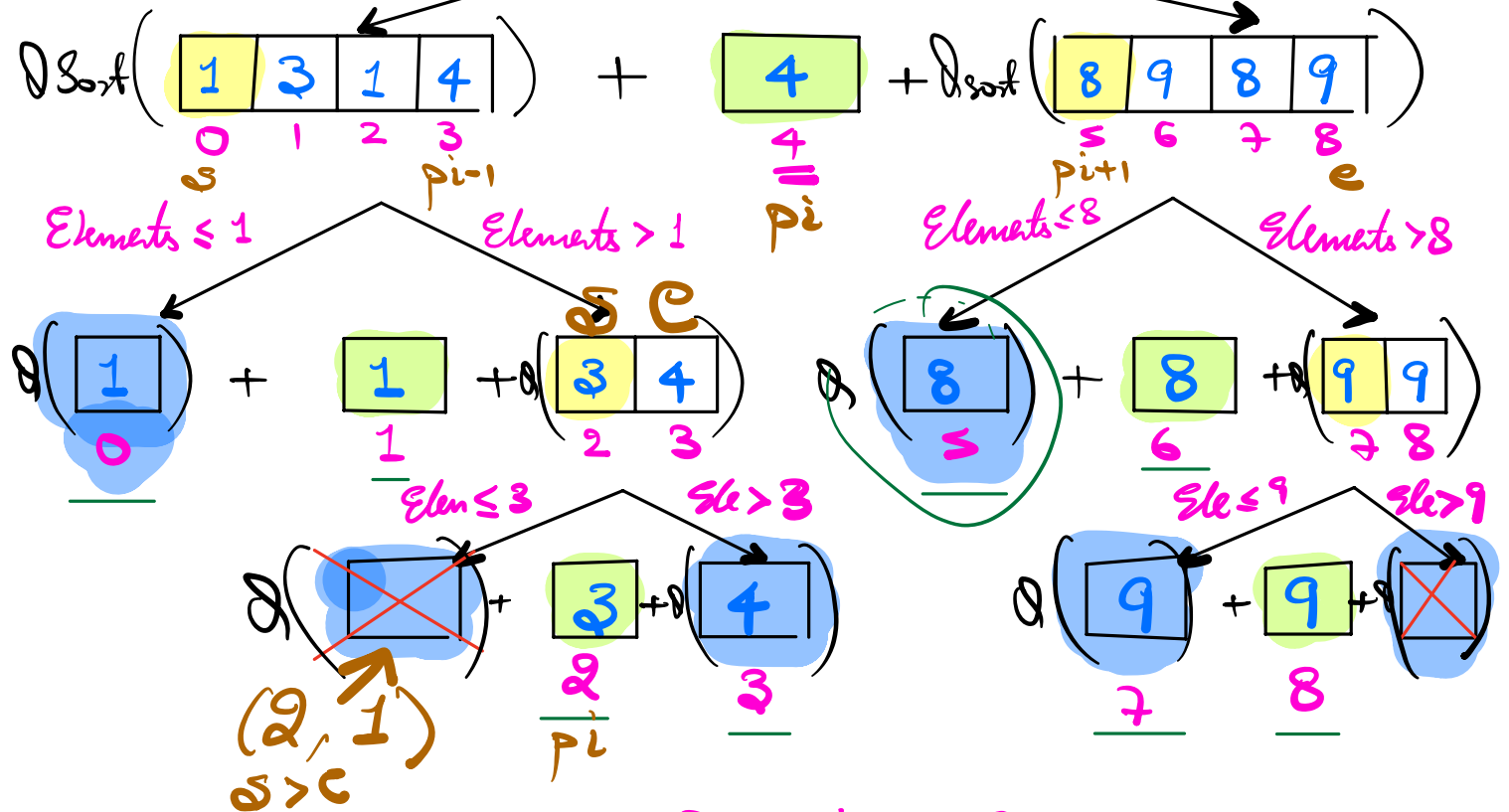
- 1) Select a pivot
- 2) Partition the array into left & right based on pivot.
- 3) Recursively call quicksort on left & right

Day Run

s								e
0	1	2	3	4	5	6	7	8
4	1	8	9	3	8	1	9	4

Elements ≤ 4

Elements > 4



O/P: $[1, 1, 3, 4, 4, 8, 8, 9, 9]$

Code

```
void quickSort(A[], s, e) {
    if (s < e) {
```

```
        int pi = partition(A, s, e);
```

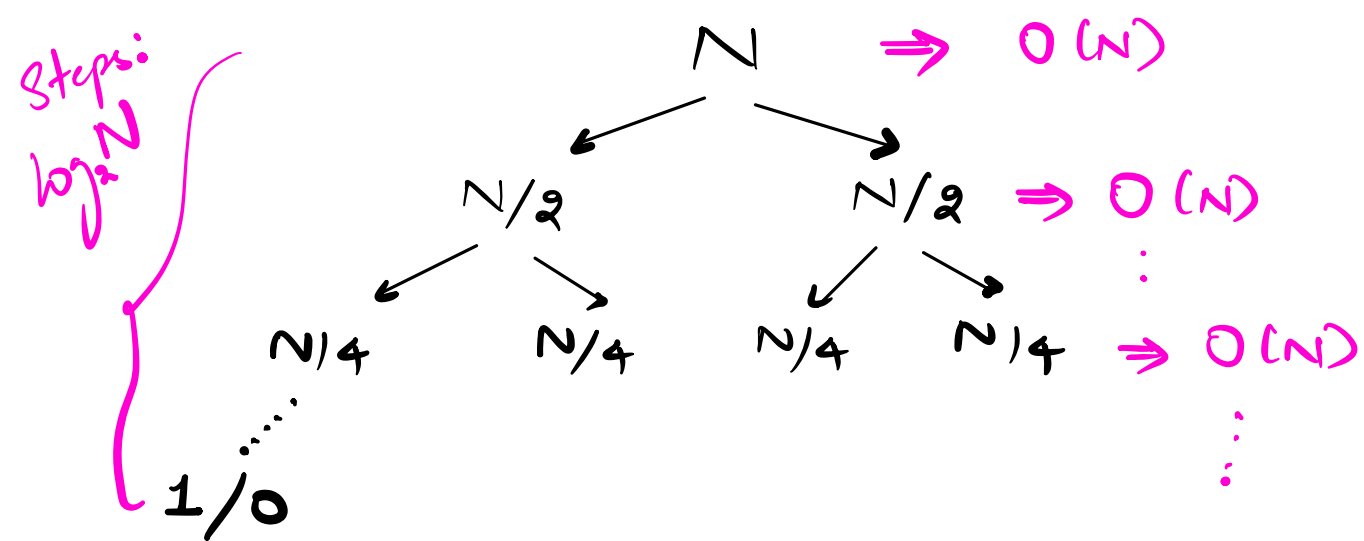
quicksort (A, s, p_i-1);

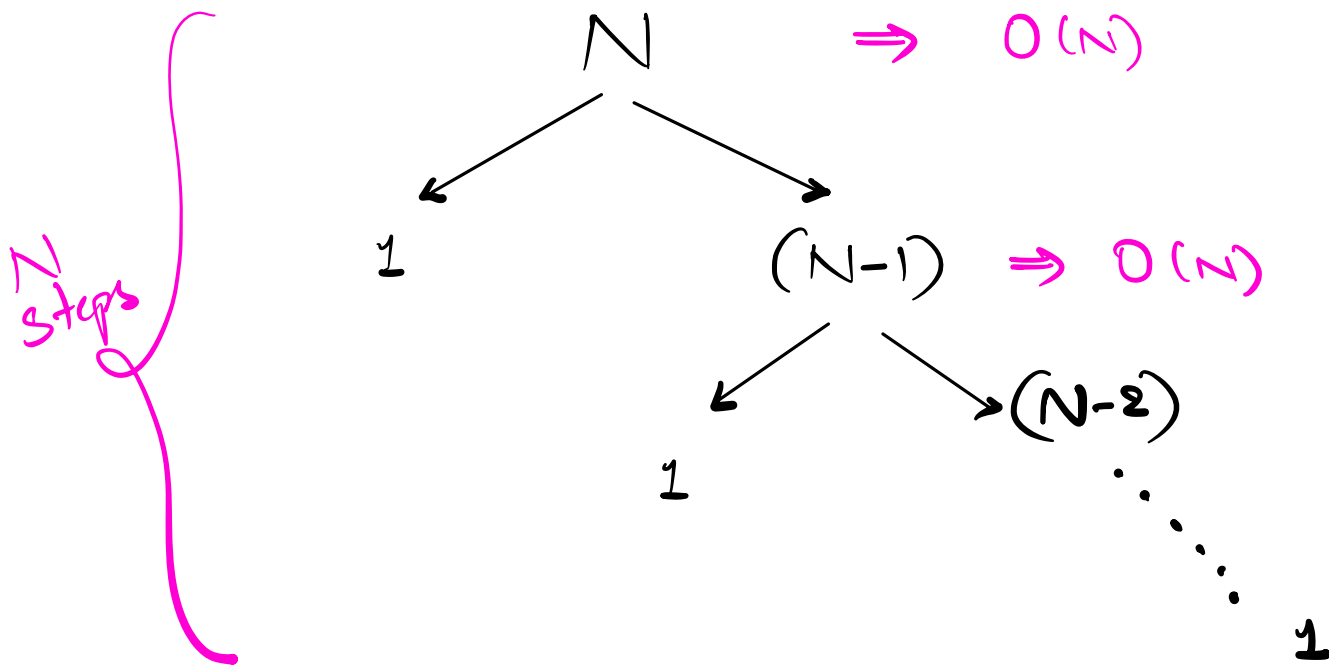
quicksort (A, p_i+1, e);

5

Time Complexity

1) Best Case





Worst Case T.C. = $O(N^2)$
 " " S.C. = $O(N)$

Quick Sort with random pivot

⇓ Observed

Quick sort performs faster than merge sort

Random pivot \Rightarrow Randomly selecting any element of the subproblem as pivot.

- Friday

- Schedule



$[0, 2, 6, 1, 0, 2, 1, 1, 2, 0]$

