

Task 1: This task is basically inversion checking for this we are using a merge sort type of algorithm. but when we are merging we are checking inversions.

Task 2: This task can be solved in linear time complexity. we declare to variables as max_sum and max_num which are float(-inf) we use a loop to find out the maximum number then we check with the other numbers using another loop.

Task 3: Task 3 is just a basic quick sort it has a time complexity of $n \log n$ in average cases

Task 4: We were told to sort this task without using partition. the task is basically quickselect, a famous problem of leetcode. we can take a pivot. here I took the last element. first we find and store smaller elements in a list and bigger in another, and using branching find out the ~~sm~~ kth smallest

Simulation

$k = 3$

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

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

pivot

as $\text{len}(\text{arr}_L) \geq k$ is true

$k = 3$

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

~~empty~~ [1, 2] [3] ~~[1, 2, 3]~~ [empty]
pivot

$\text{len}(\text{arr}_L) = k - 1 = \text{true}$

\therefore Ans is 3