

Sorting

Input: a set of n elements
 a_1, a_2, \dots, a_n
numbers

Output: A permutation of the numbers such that

$$a_{\pi(1)} \leq a_{\pi(2)} \leq \dots \leq a_{\pi(n)}$$

either they are non-decreasing or non-increasing.

Ex^m

8 5 1 2 6 4



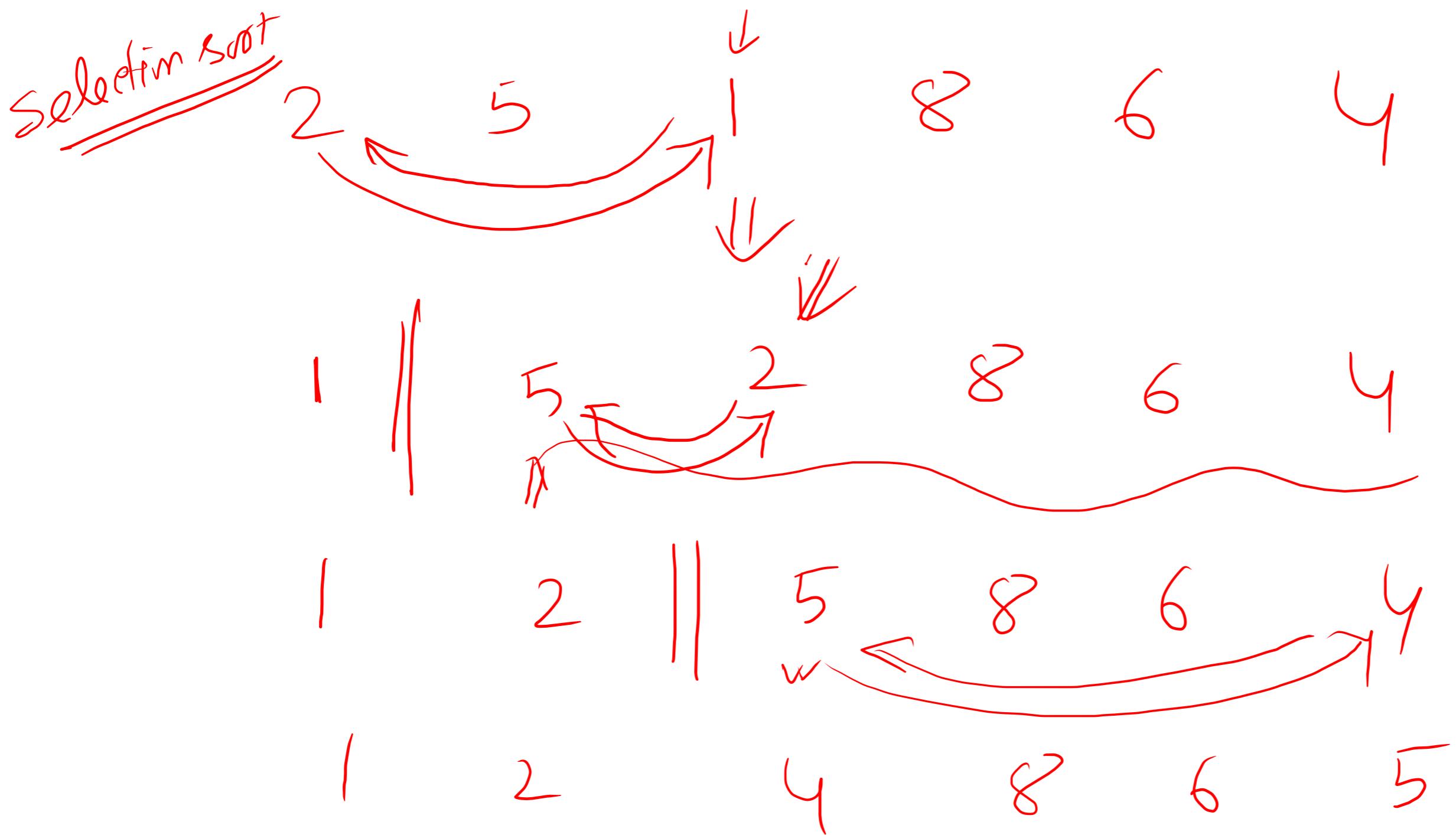
1 2 4 5 6 8

A simple algorithm

- consider all possible permutations of the n elements
- Return the desired one - $O(n)$ $n!$

Total time: $O(n n!)$

Can we do better?



Running time: $n + (n-1) + \dots + 1 = \frac{n(n+1)}{2} = O(n^2)$

Bubble sort

8 4 2 5 9 6

compare and the
maximum will be
the right one.

4 8 2 5 9 6

$$= \frac{n(n-1)}{2}$$

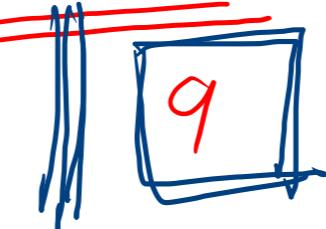
4 2 8 5 9 6

$$= O(n^2)$$

4 2 5 8 9 6

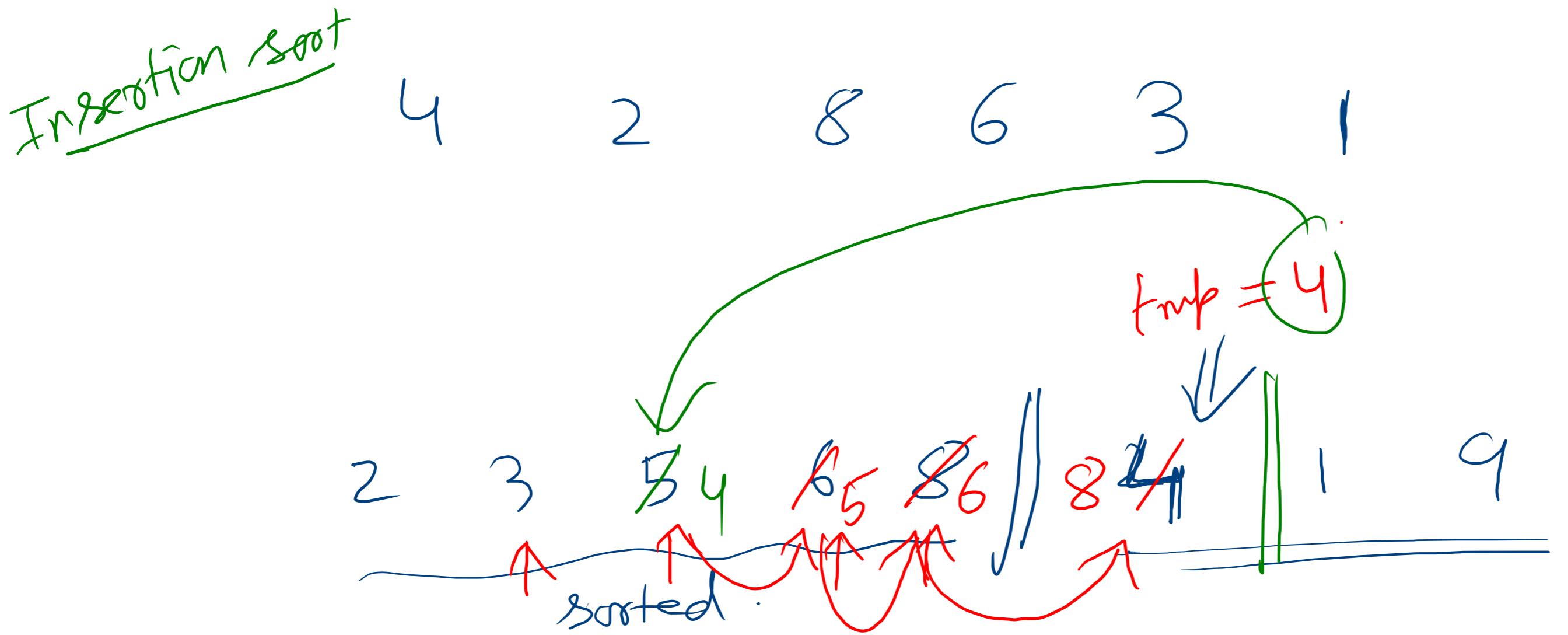
4 2 5 8 9 6

4 2 5 8 9 6



Running time:

$$(n-1) + (n-2) + \dots + 1$$



Insertion-Sort(A, n)

for $j = 1$ to $n-1$

$\text{tmp} = A[j]$

$i = j - 1$

while $i > 0$ and $A[i] > \text{tmp}$

$A[i+1] = A[i]$

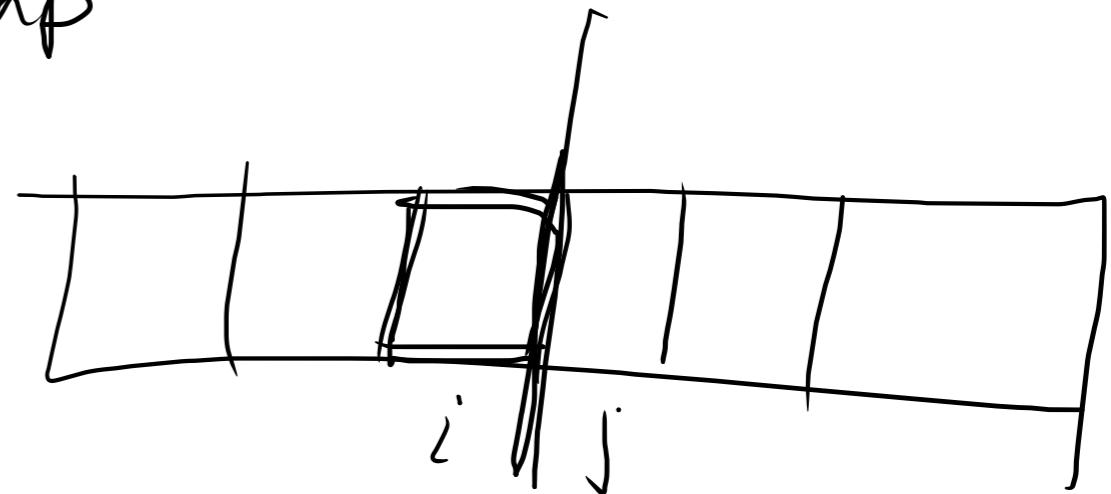
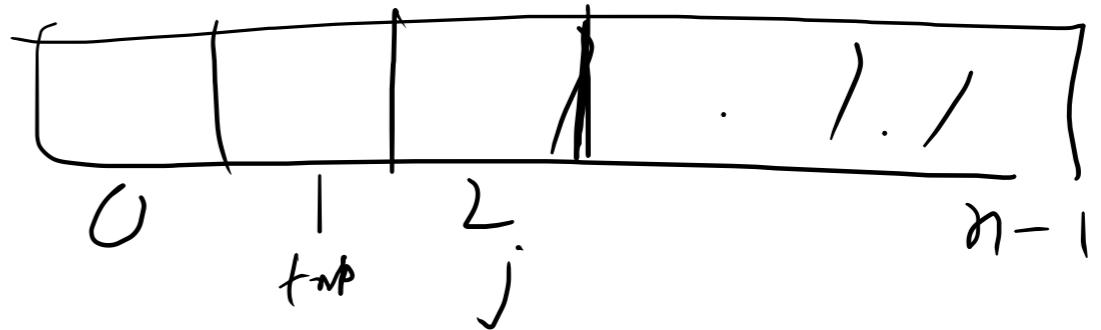
$i = i - 1$

$A[i+1] = \text{tmp}$

Time

$$1 + 2 + 3 + \dots + n-1$$

$$\approx O(n^2)$$

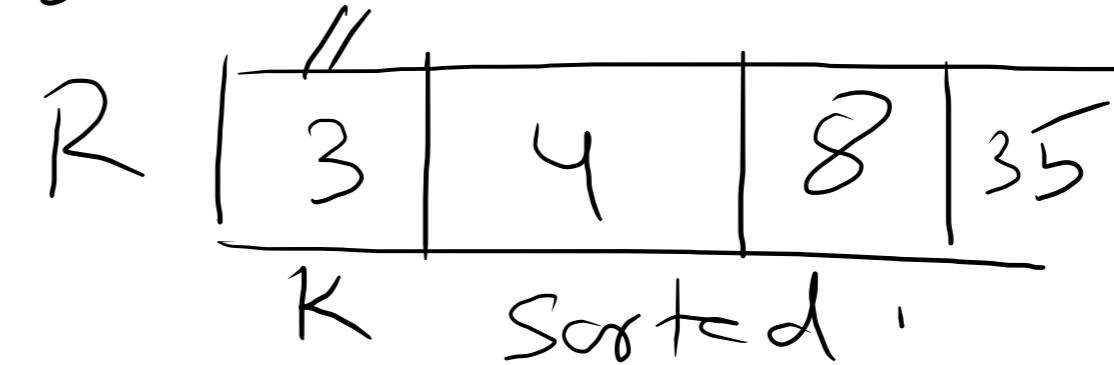
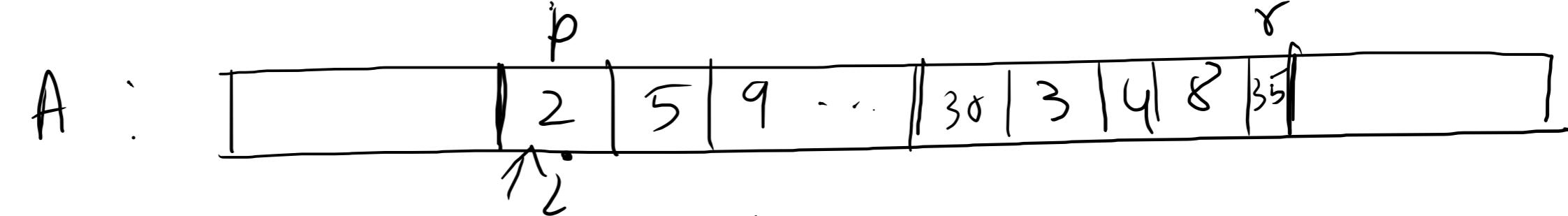


Inplace Sorting:

Apart from input only a constant amount of storage is required.

Otherwise the sorting is not an inplace sorting.

Merge sort



Target: want to sort the element in $L \cup R$.

Compare first element of $L \cup R$ } minimum goes to i th position.

only the minimum element's index gets incremented.

\downarrow
 i gets incremented.

L	<table border="1"> <tr> <td>2</td><td>5</td><td>9</td><td>12</td><td>15</td><td>20</td><td>30</td></tr> </table>	2	5	9	12	15	20	30	R	<table border="1"> <tr> <td>3</td><td>4</td><td>8</td><td>35</td></tr> </table>	3	4	8	35
2	5	9	12	15	20	30								
3	4	8	35											
A	<table border="1"> <tr> <td></td><td>2</td><td>3</td><td>4</td><td>58</td><td>9</td><td>12</td><td>15</td><td>20</td><td>30</td><td>35</td></tr> </table>		2	3	4	58	9	12	15	20	30	35		
	2	3	4	58	9	12	15	20	30	35				

Diagram illustrating a merge operation between two sorted lists, L and R, to form a new list A.

The lists are represented as arrays:

- List L:** [2, 5, 9, 12, 15, 20, 30]
- List R:** [3, 4, 8, 35]
- List A (Result):** [2, 3, 4, 58, 9, 12, 15, 20, 30, 35]

Red annotations indicate the merging process:

- Index i:** Points to the current element being compared from List L. It starts at index 0 and moves right through index 6.
- Index j:** Points to the current element being compared from List R. It starts at index 0 and moves right through index 3.
- Index k:** Points to the current element being inserted into List A. It starts at index 0 and moves right through index 9.
- Red X marks:** Indicate elements that have been processed or removed from their respective lists. In List L, '5' and '12' are crossed out. In List R, '3' and '8' are crossed out.