

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

\Downarrow

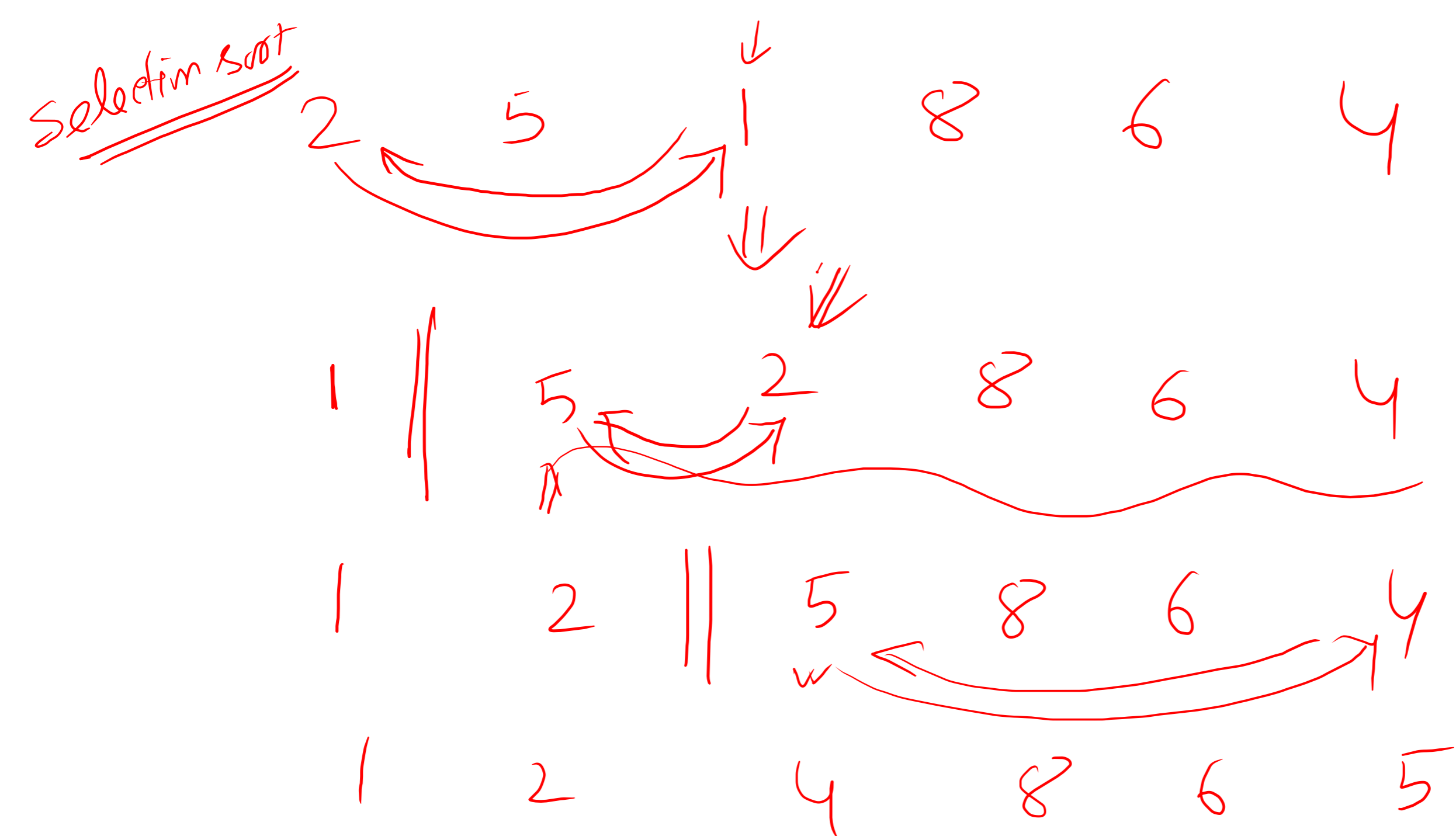
1 2 4 5 6 8

A simple algorithm

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

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

4 2 8 5 9 6

4 2 5 8 9 6

4 2 5 8 9 6

4 2 5 8

6 9

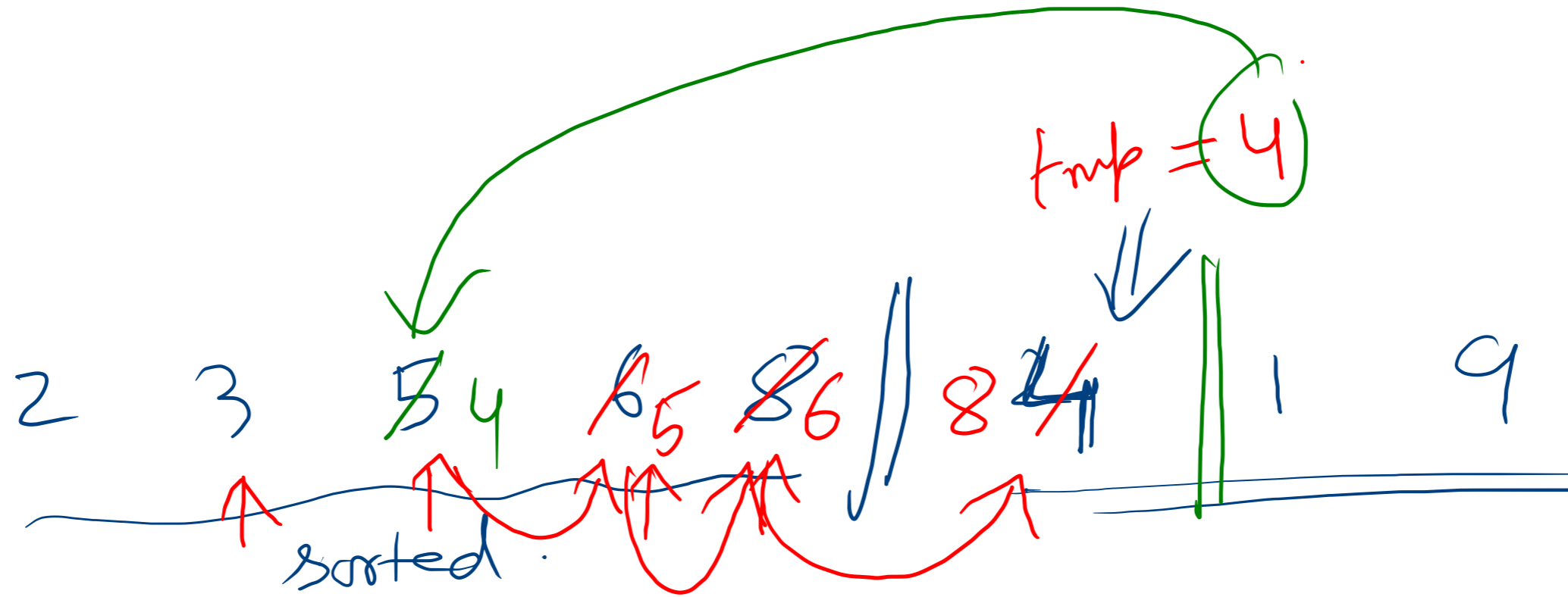
Running time:
 $(n-1) + (n-2) + \dots + 1$

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

$$= O(n^2)$$

Insertion sort

4 2 8 6 3 1



Insertion-sort(A, n)

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

$tmp = A[j]$

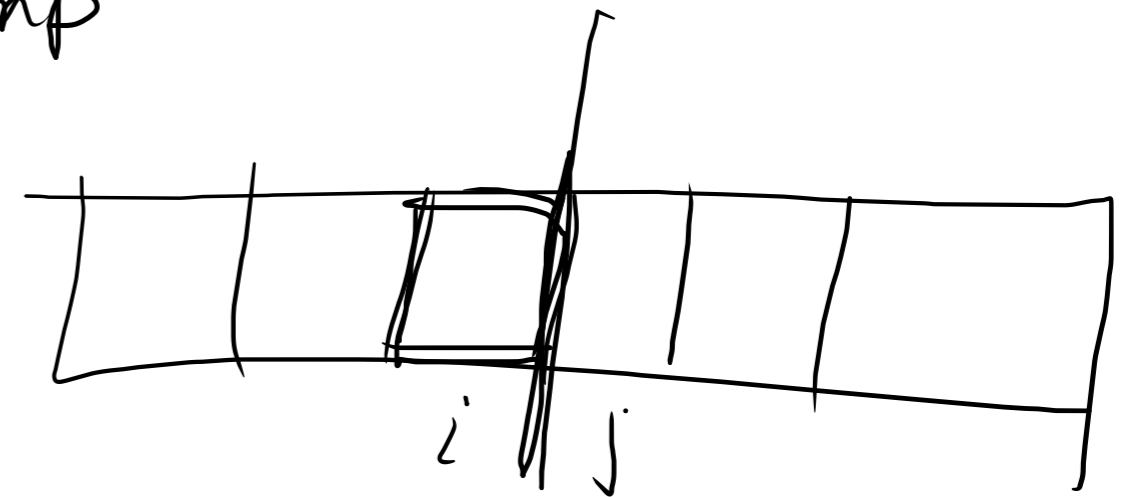
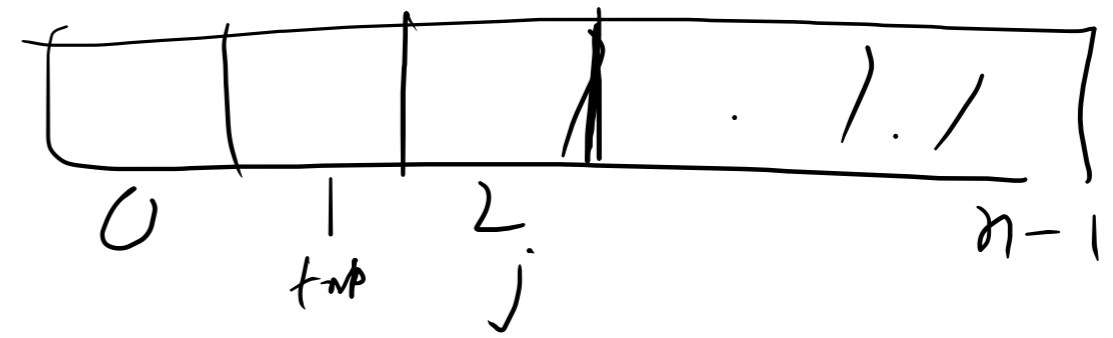
$i = j-1$

 while $i > 0$ and $A[i] > tmp$

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

$i = i-1$

$A[i+1] = tmp$



Time

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

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

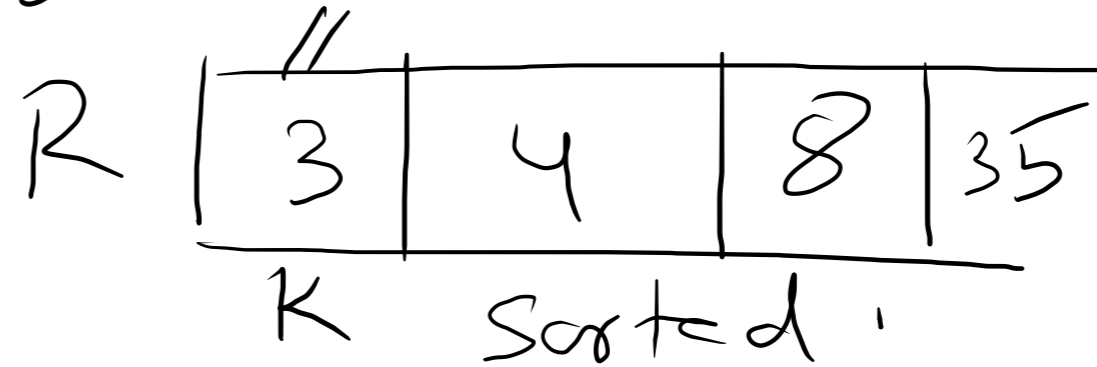
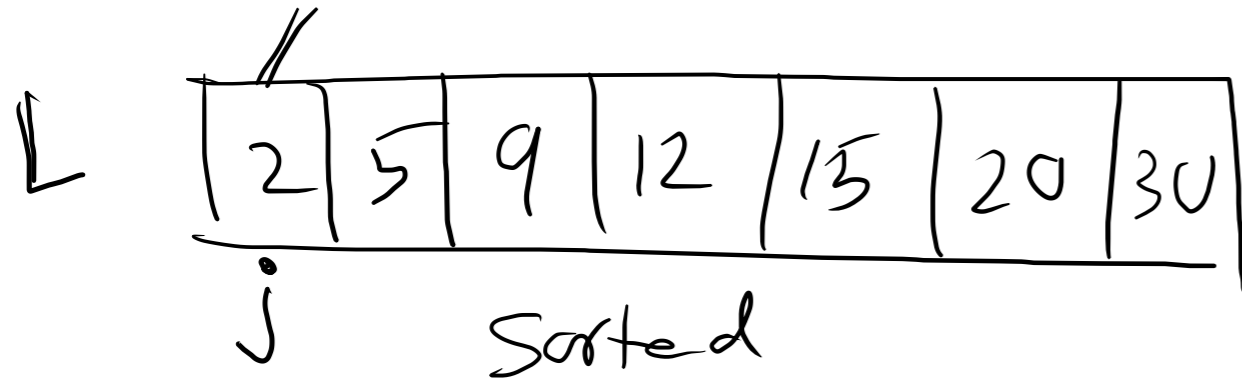
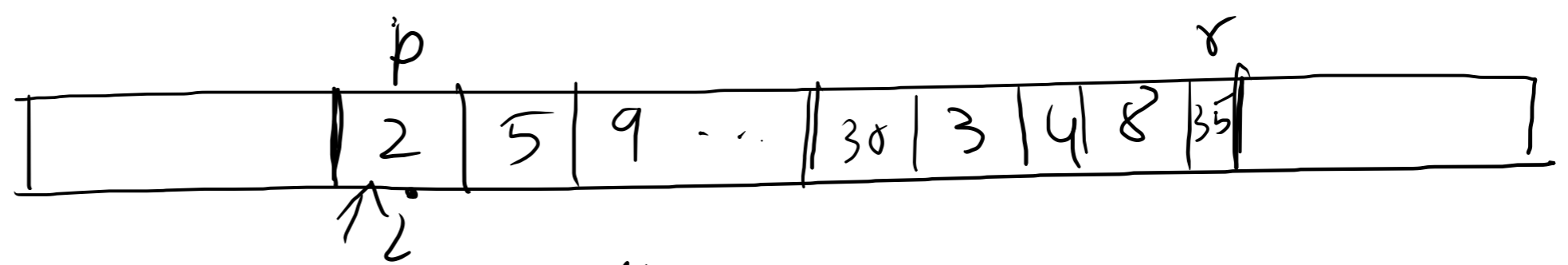
Inplace Sorting:

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

Otherwise the sorting is not an inplace sorting.

Merge Sort

A :



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

compare first element of L }
" " of R } minimum goes to i-th
position.

only the minimum
element's index gets
incremented.

↓
i gets incremented.

L

2	5	9	12	15	20	30
---	---	---	----	----	----	----

~~j~~ ~~j~~ ~~j~~ ~~j~~ j

R

3	4	8	35
---	---	---	----

~~K~~ ~~K~~ ~~K~~ ~~K~~

A

	2	3	4	5	8	9	12	15	20	30	35
--	---	---	---	---	---	---	----	----	----	----	----

~~j~~ ~~j~~ ~~j~~ ~~j~~ ~~j~~ j j