Divide and conquer

Sorting algorithms

Sorting: Given a sequence of numbers

$$a_1, a_2, \ldots, a_n$$

output: Return a permutation of the numbers such that

$$a_{i_1} \leq a_{i_2} \leq \ldots \leq a_{i_n}$$

Ex

Input: 2 1 6 3 9

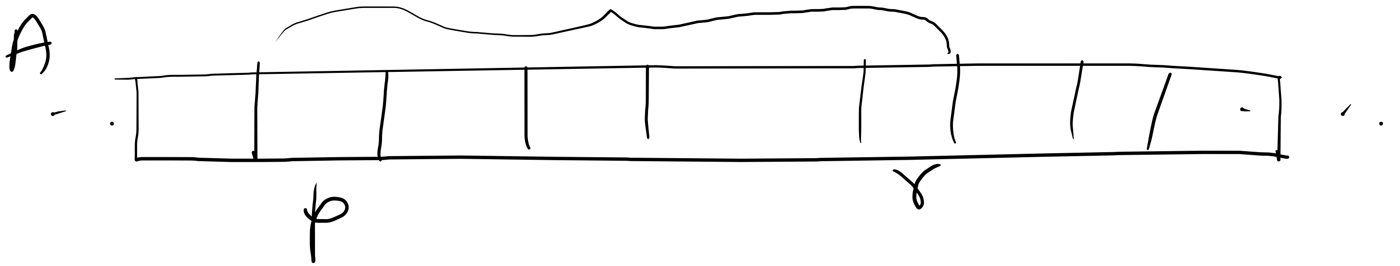
output: 1 2 3 6 9

Merge Sort

High-level idea

- Divide the array (A) into roughly two equal partitions - L and R
- sort the array L
- sort the array R
- merge the two sorted arrays L & R to get the sorted array A.

$\text{Mergesort}(A, p, r)$ — $T(n)$



if $p < r$ —— $\theta(1)$

$q = \frac{p+r}{2}$ —— $\theta(1)$

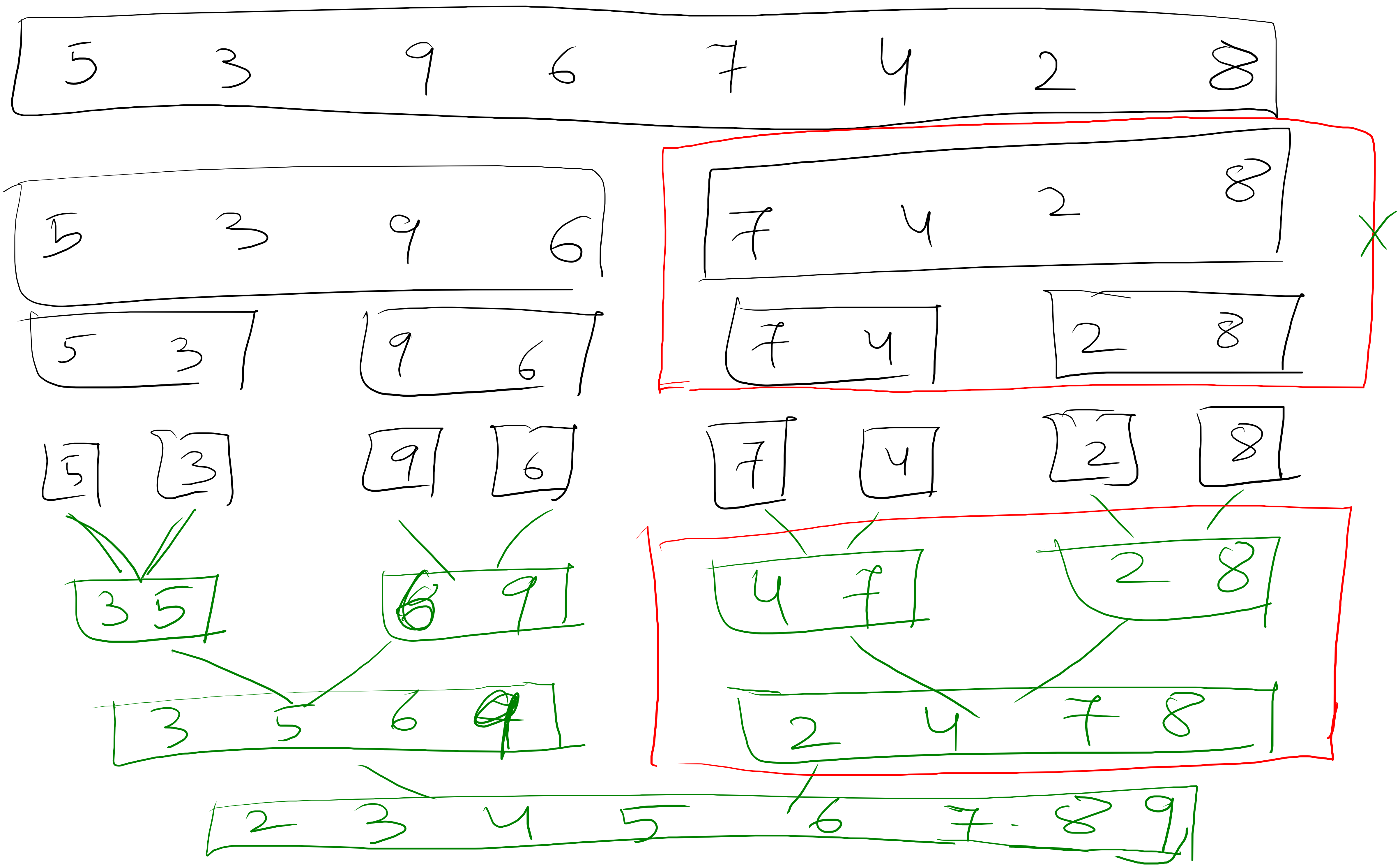
$\text{mergesort}(A, p, q)$ — $T(\frac{n}{2})$

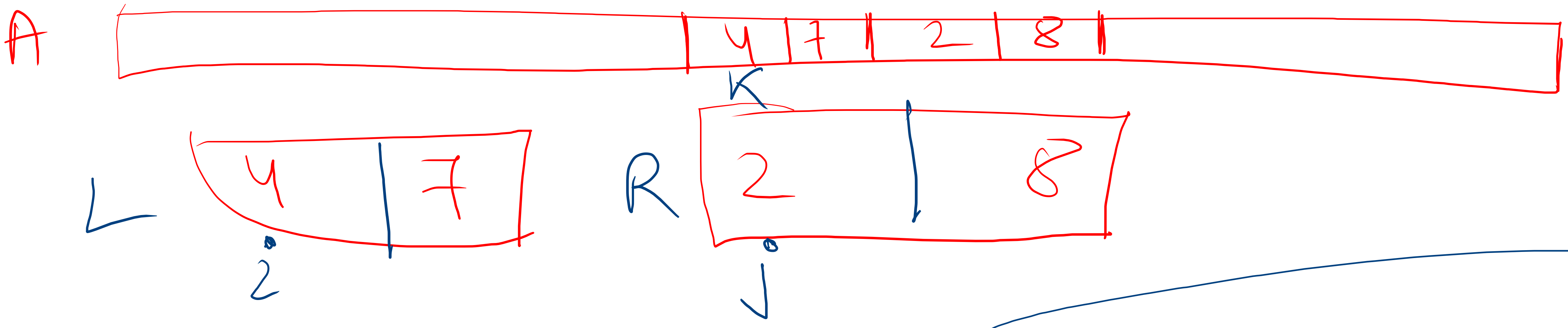
$\text{mergesort}(A, q+1, r)$ —— $T(n/2)$

$\text{merge}(A, p, q, r)$ —— $\theta(n)$

Total time: $T(n) = 2T(\frac{n}{2}) + \theta(n)$

$$\Rightarrow T(n) = \theta(n \log n)$$


5 3 9 6 7 4 2 8
5 3 9 6
7 4 2 8
5 3
9 6
7 4
2 8
5
3
9
6
7
4
2
8
3 5
6 9
4 7
2 8
3 5 6 9
2 4 7 8
2 3 4 5 6 7 8 9



if $L[i] \le R[j]$

$A[k] \leftarrow L[i]$

$k = k + 1$

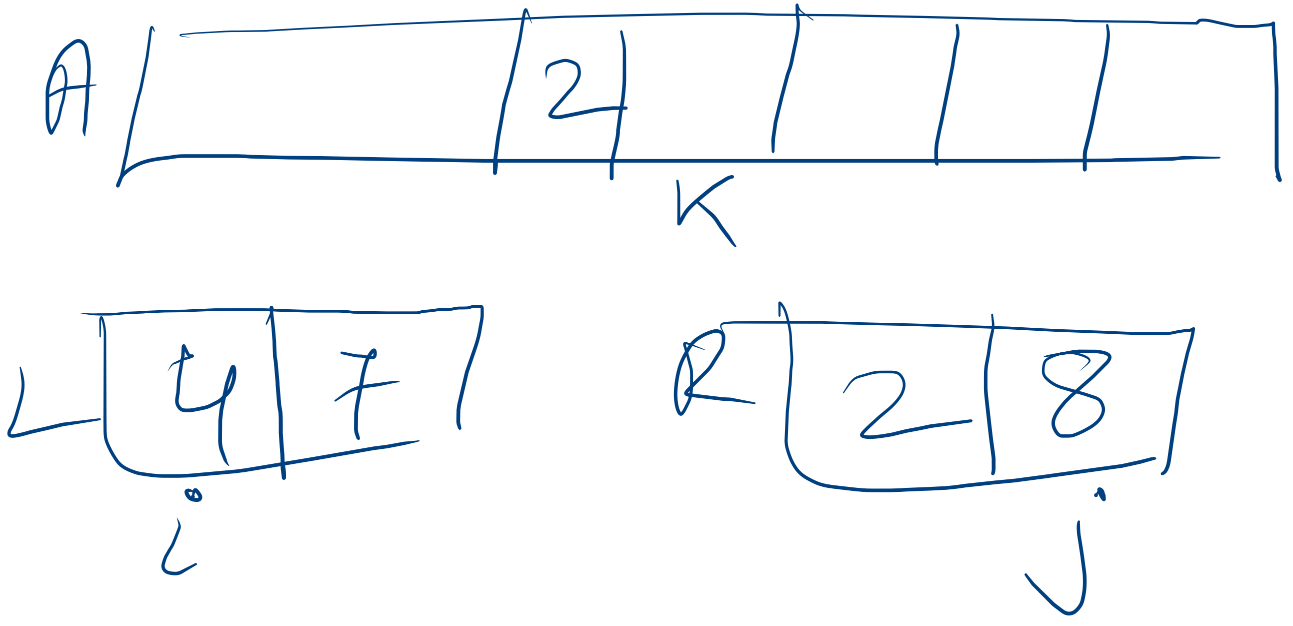
$i = i + 1$

else

$A[k] \leftarrow R[j]$

$k = k + 1$

$j = j + 1$



```
merge(A, p, q, r)
  n1 ← q - p + 1
  n2 ← r - q
  for i = 1 to n1
      L[i] ← A[p+i-1]
  for j = 1 to n2
      R[j] ← A[q+j]
  L[n1+1] ← ∞
  R[n2+1] ← ∞
⟹ for k = p to r
      if L[i] ≤ R[j]
          A[k] ← L[i]
          i = i+1
      else A[k] ← R[j]
          j = j+1
```

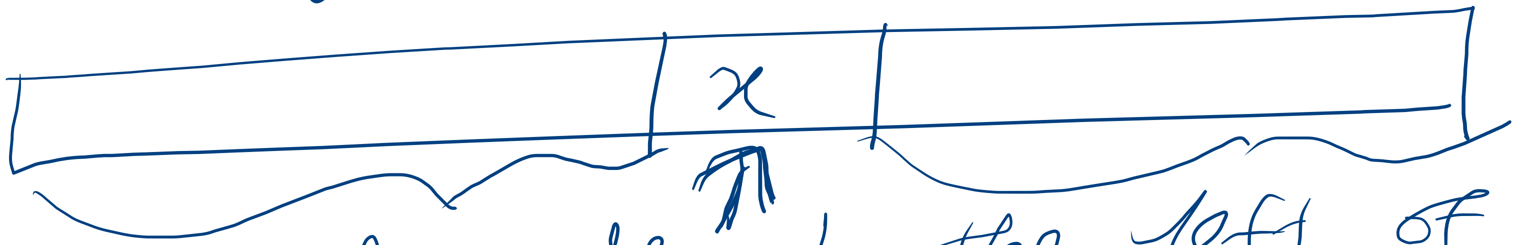
running time :- $O(n)$

At the start of each iteration of the for loop. subarray $A[p \ldots k-1]$ contains the $k-p$ sorted elements of L and R and $L[i]$ and $R[j]$ are the smallest elements of L and L that are yet to copied in A.

Quicksort

highlevel idea

- we choose a suitable element x
- based on the element x, we partition the array into two parts



- all the elements to the left of x are $\leq x$
- all the elements to the right of x are $> x$

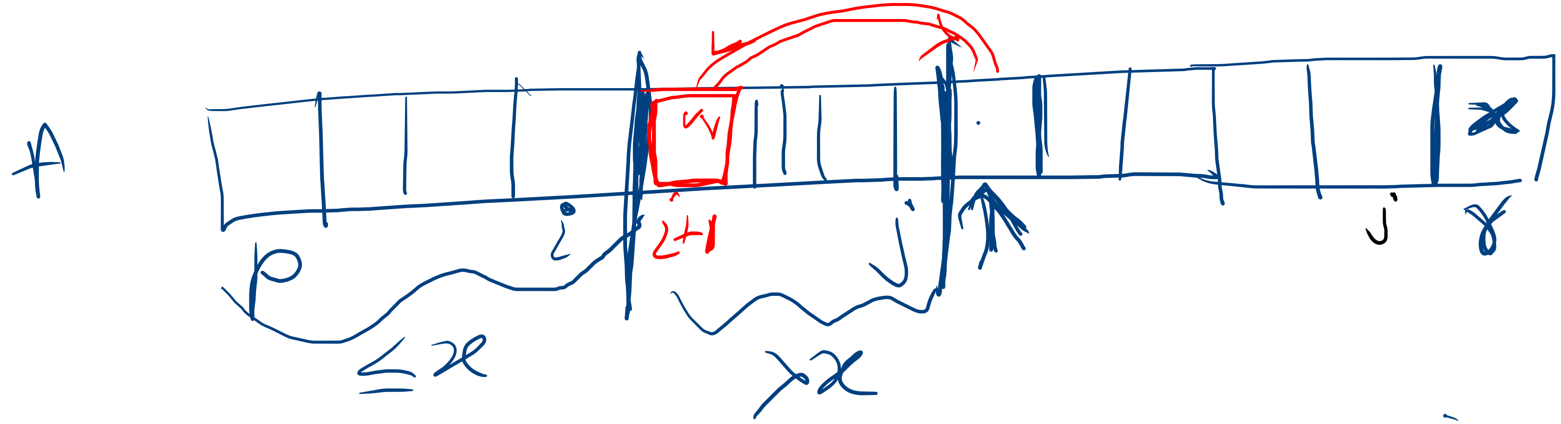
Quicksort (A, p, r) —— $T(n)$

if p < r —— $\theta(1)$

q = partition (A, p, r) ——→ $\theta(n)$

Quicksort (A, p, q-1) ——

Quicksort (A, q+1, r) ——

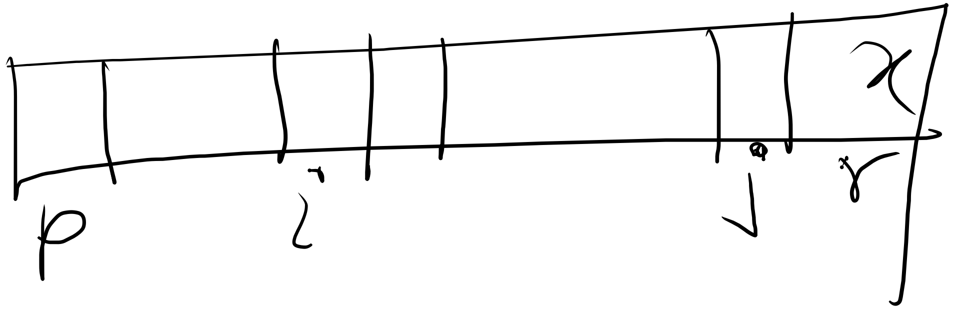


$A[j+1]$ compares with x.

either $A[j+1] \le x$

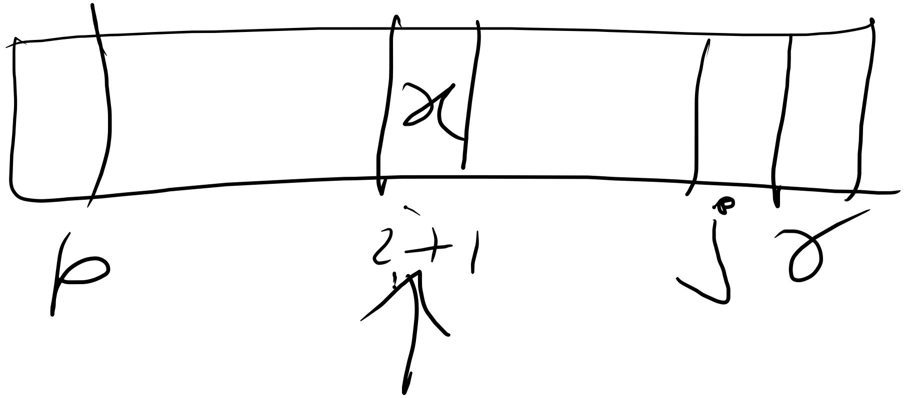
$i = i+1$

swap $A[j+1] \leftrightarrow A[i]$

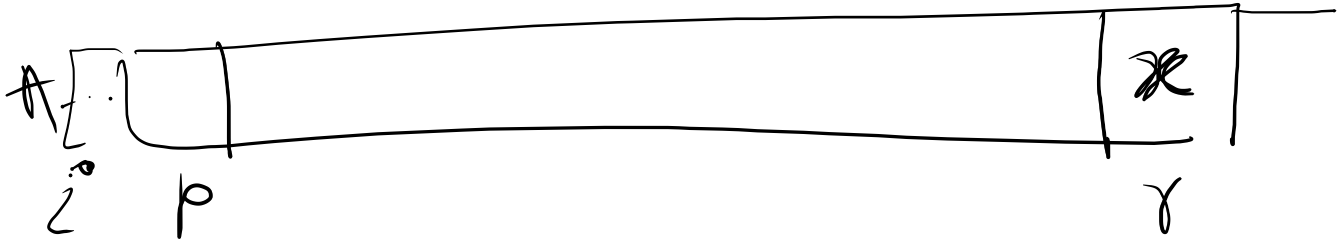


$A[j+1] > x$

$j = j+1$



partition(A, p, r)
  x = A[r]
  i = p - 1
  for j = p to r - 1
      if A[j] ≤ x
          i = i + 1
          swap A[i] ⟺ A[j]
  swap A[i+1] ⟺ A[r]
  return i+1



All the elements are same.