

# Probabilistic Analysis & Quicksort

CS3026 – Analysis & Design of Algorithms

Angel Napa



# Índice

1. Random Variables
2. Probabilistic Analysis
3. Quicksort

# 2

## Probabilistic Analysis

### HIRE-ASSISTANT( $n$ )

```
1  best = 0           // candidate 0 is a least-qualified dummy candidate
2  for  $i = 1$  to  $n$ 
3      interview candidate  $i$ 
4      if candidate  $i$  is better than candidate best
5          best =  $i$ 
6      hire candidate  $i$ 
```

**Figure 1:** Cormen, Introduction to Algorithms



# 3 Quicksort



# Algorithm

QUICKSORT( $A, p, r$ )

```
1  if  $p < r$ 
2      // Partition the subarray around the pivot, which ends up in  $A[q]$ .
3       $q = \text{PARTITION}(A, p, r)$ 
4      QUICKSORT( $A, p, q - 1$ ) // recursively sort the low side
5      QUICKSORT( $A, q + 1, r$ ) // recursively sort the high side
```

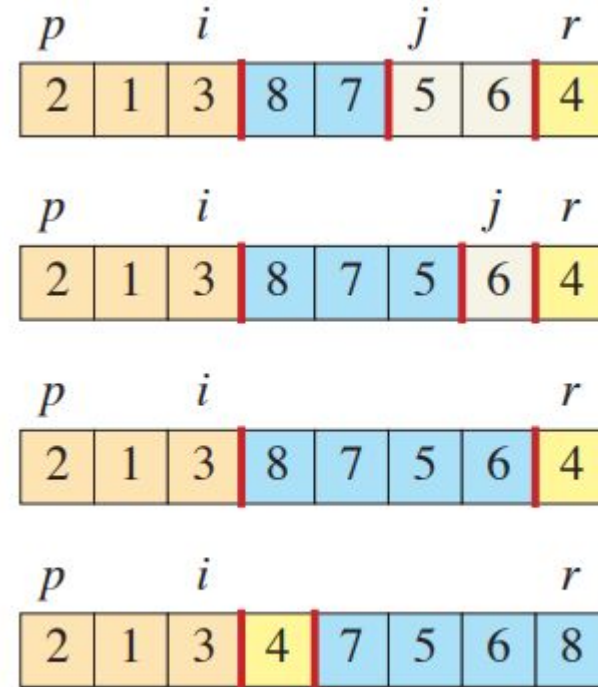
# Partition

PARTITION( $A, p, r$ )

```
1   $x = A[r]$  // the pivot
2   $i = p - 1$  // highest index into the low side
3  for  $j = p$  to  $r - 1$  // process each element other than the pivot
4      if  $A[j] \leq x$  // does this element belong on the low side?
5           $i = i + 1$  // index of a new slot in the low side
6          exchange  $A[i]$  with  $A[j]$  // put this element there
7  exchange  $A[i + 1]$  with  $A[r]$  // pivot goes just to the right of the low side
8  return  $i + 1$  // new index of the pivot
```

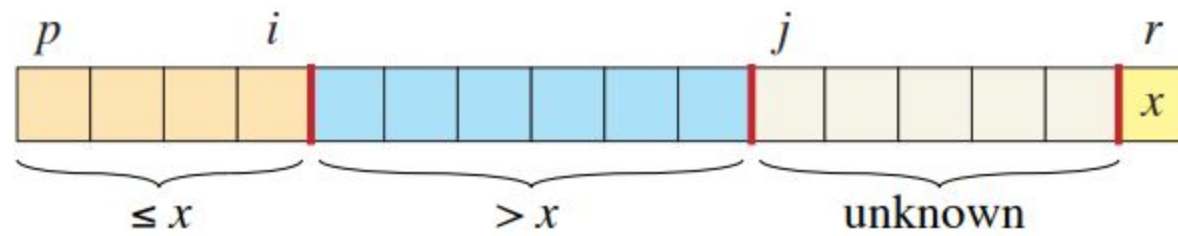
Diagram illustrating the steps of the partitioning process in the quicksort algorithm. The array is  $[2, 8, 7, 1, 3, 5, 6, 4]$ . The pivot is 4 (at index 7). The partitioning process moves elements less than the pivot to the left and elements greater than the pivot to the right. The steps shown are:

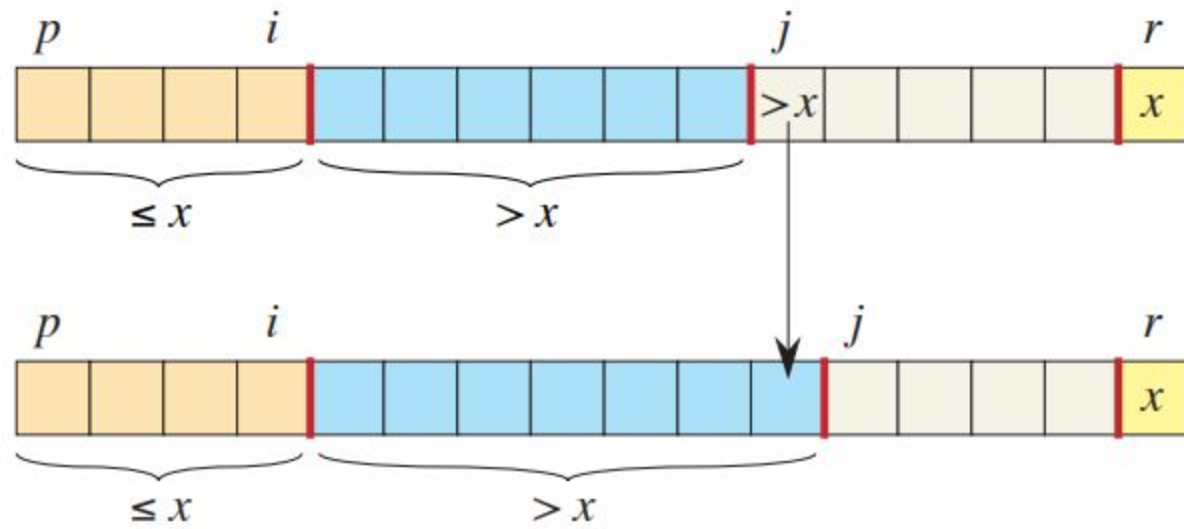
- Initial state:  $i = 0, p = 0, j = 7, r = 7$ . Array:  $[2, 8, 7, 1, 3, 5, 6, 4]$ .
- Step 1:  $i = 0, p = 0, j = 7, r = 7$ . Array:  $[2, 8, 7, 1, 3, 5, 6, 4]$ . Element 2 is less than 4, so it stays on the left.
- Step 2:  $i = 0, p = 0, j = 7, r = 7$ . Array:  $[2, 8, 7, 1, 3, 5, 6, 4]$ . Element 8 is greater than 4, so it is swapped with 7.
- Step 3:  $i = 0, p = 0, j = 7, r = 7$ . Array:  $[2, 7, 8, 1, 3, 5, 6, 4]$ . Element 7 is greater than 4, so it is swapped with 1.
- Step 4:  $i = 0, p = 0, j = 7, r = 7$ . Array:  $[2, 1, 8, 7, 3, 5, 6, 4]$ . Element 1 is less than 4, so it stays on the left.
- Step 5:  $i = 0, p = 0, j = 7, r = 7$ . Array:  $[2, 1, 8, 7, 3, 5, 6, 4]$ . Element 3 is less than 4, so it stays on the left.
- Step 6:  $i = 0, p = 0, j = 7, r = 7$ . Array:  $[2, 1, 8, 7, 3, 5, 6, 4]$ . Element 5 is greater than 4, so it is swapped with 6.
- Step 7:  $i = 0, p = 0, j = 7, r = 7$ . Array:  $[2, 1, 8, 7, 3, 6, 5, 4]$ . Element 6 is greater than 4, so it is swapped with 4.
- Final state:  $i = 0, p = 0, j = 7, r = 7$ . Array:  $[2, 1, 7, 8, 3, 5, 6, 4]$ .



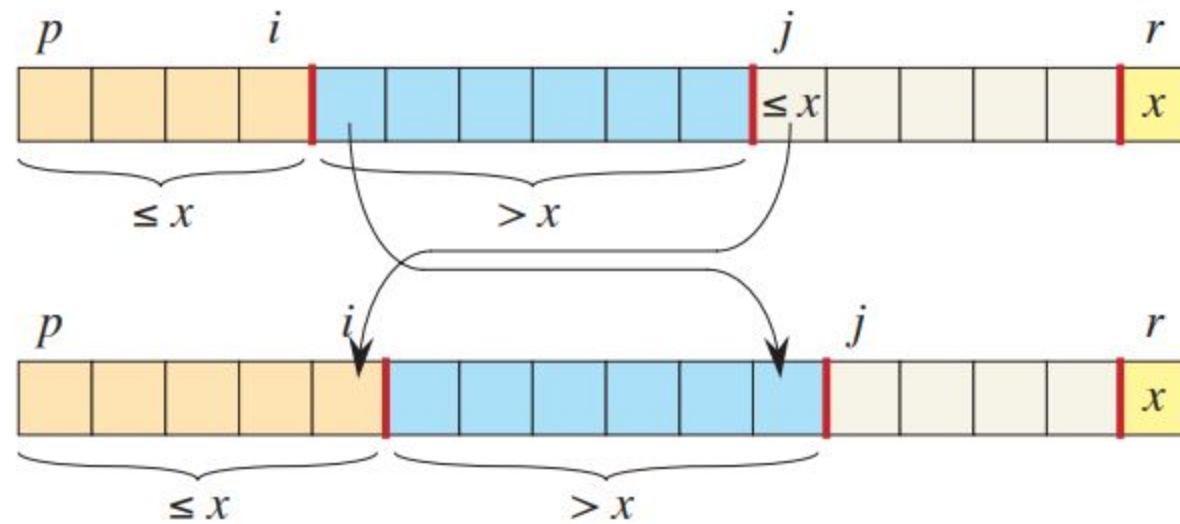


# Invariant



$A[j] > x$ 

$$A[j] \leq x$$



# General Formula

$$T(n) = \max \{T(q) + T(n - 1 - q) : 0 \leq q \leq n - 1\} + \Theta(n)$$

# Visualization

$$T(n) \geq \min_{0 \leq q \leq n-1} \{T(q) + T(n-1-q)\} + \Omega(n).$$



# Thank you!

*Let's keep finding mistakes!*

