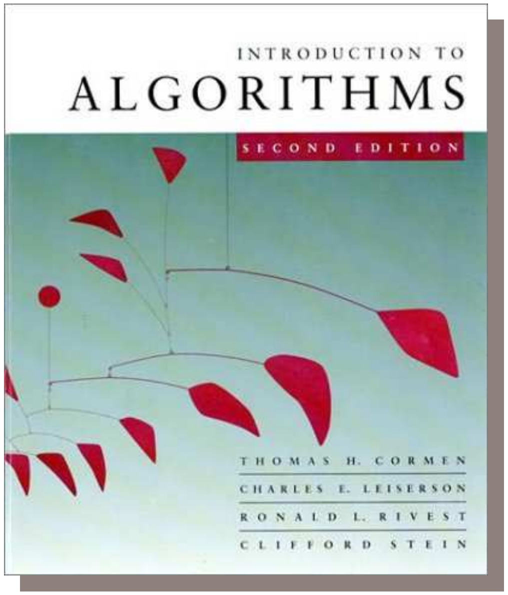
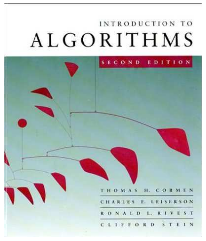
Data Structures and Algorithms
CSE 465



LECTURE 7
Quick Sort

Sofya Raskhodnikova and Adam Smith



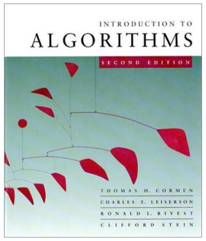
Review questions

- Use the master theorem to solve the following recurrences:
  - $T(n) = 2\,T(n/4) + 5n^{1/4}$

    (Answer: $\Theta(n^{1/2})$)
  - $T(n) = 2\,T(n/4) + n^{1/2}$

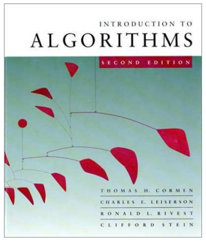
    (Answer: $\Theta(n^{1/2}\log n)$)
  - $T(n) = 2\,T(n/4) + n$

    (Answer: $\Theta(n)$)



Quicksort

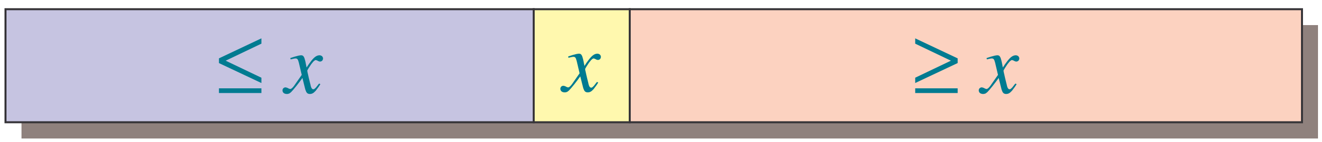
- Proposed by C.A.R. Hoare in 1962.
- Divide-and-conquer algorithm.
- Sorts "in place" (like insertion sort, but not like merge sort).
- Very practical (with tuning).



Divide and conquer

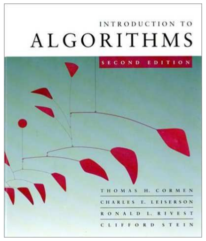
Quicksort an n-element array:

1. ***Divide:*** Partition the array into two subarrays around a ***pivot*** x such that elements in lower subarray $\leq x \leq$ elements in upper subarray.



2. ***Conquer:*** Recursively sort the two subarrays.
3. ***Combine:*** Nothing.

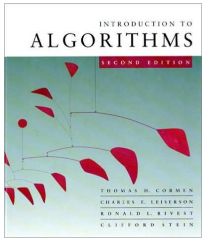
Key: *Linear-time partitioning subroutine.*



Partitioning subroutine

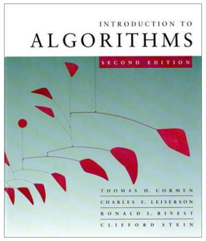
```
PARTITION(A, p, q)  ▷ A[p . . q]
    x ← A[p]        ▷ pivot = A[p]
    i ← p
    for j ← p + 1 to q
        do if A[j] ≤ x
              then  i ← i + 1
                    exchange A[i] ↔ A[j]
    exchange A[p] ↔ A[i]
    return i
```

Running time $= O(n)$ for n elements.

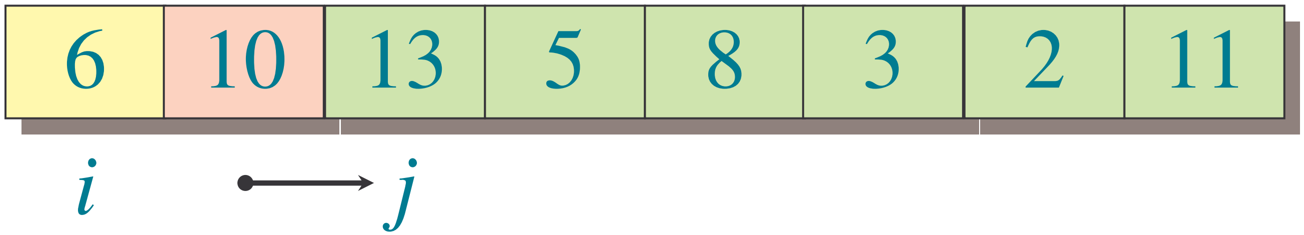


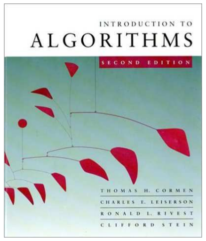
Example of partitioning

6	10	13	5	8	3	2	11
i	*j*						

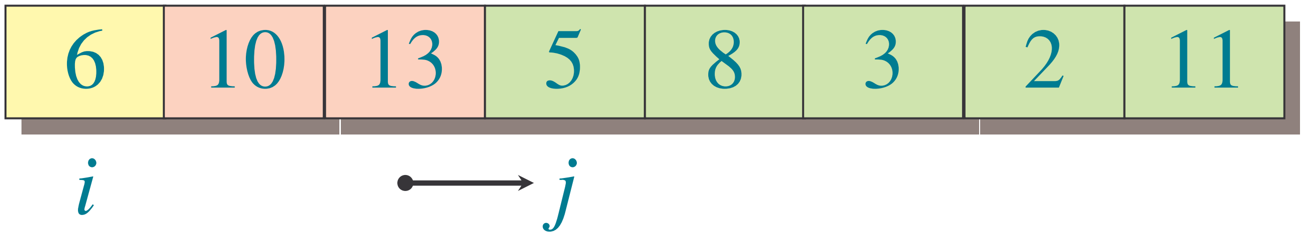


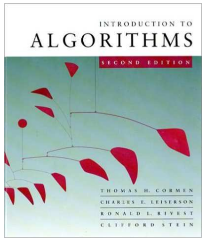
Example of partitioning



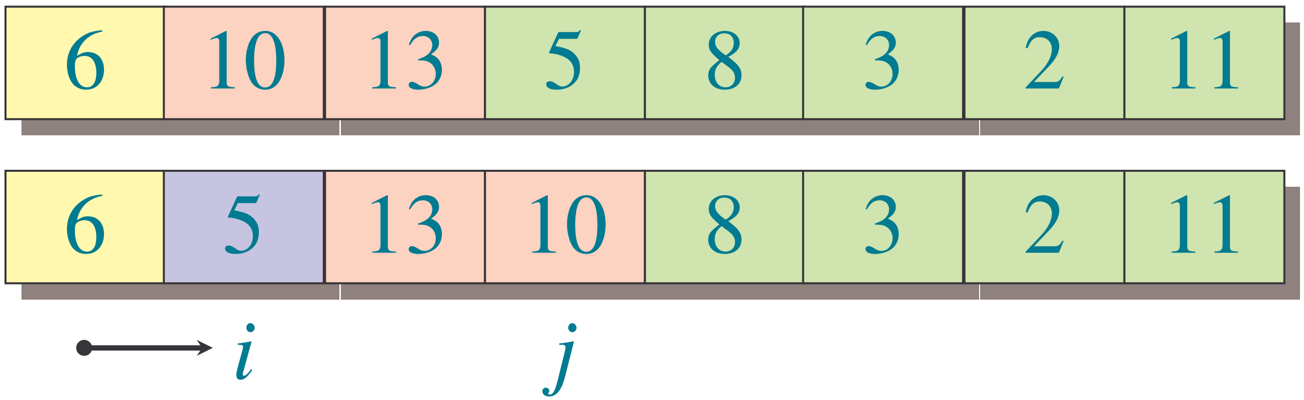


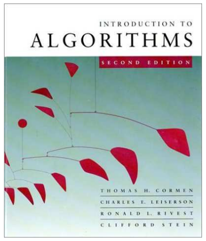
Example of partitioning



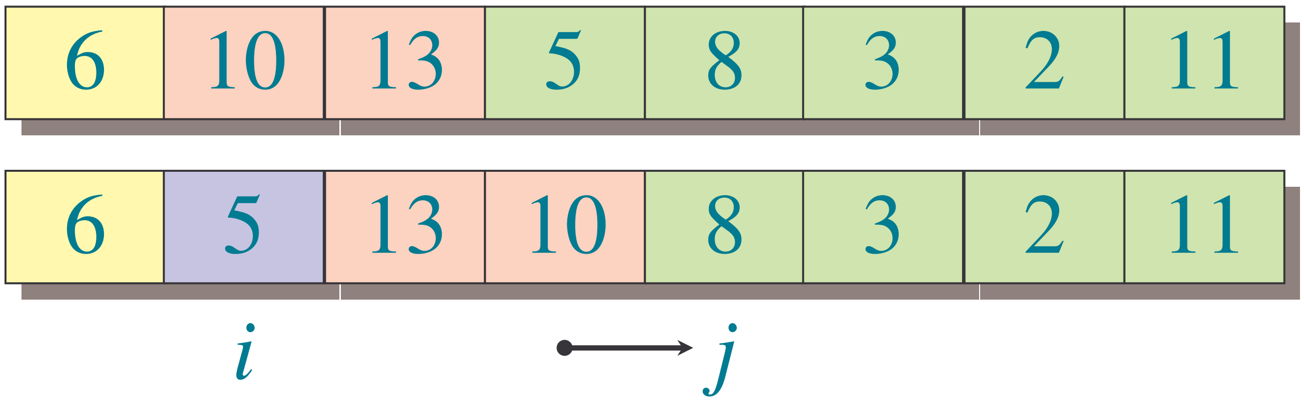


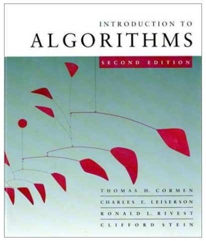
Example of partitioning



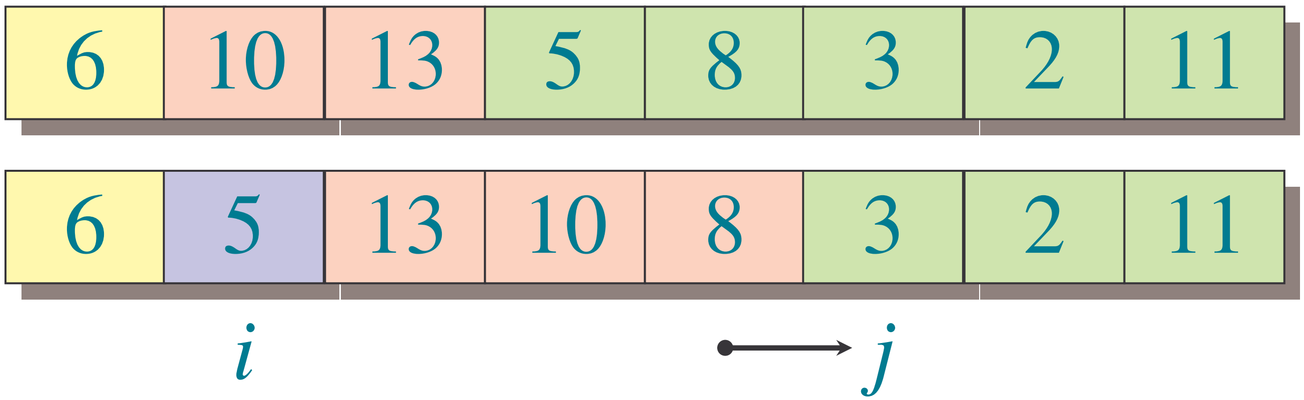


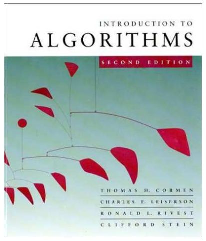
Example of partitioning



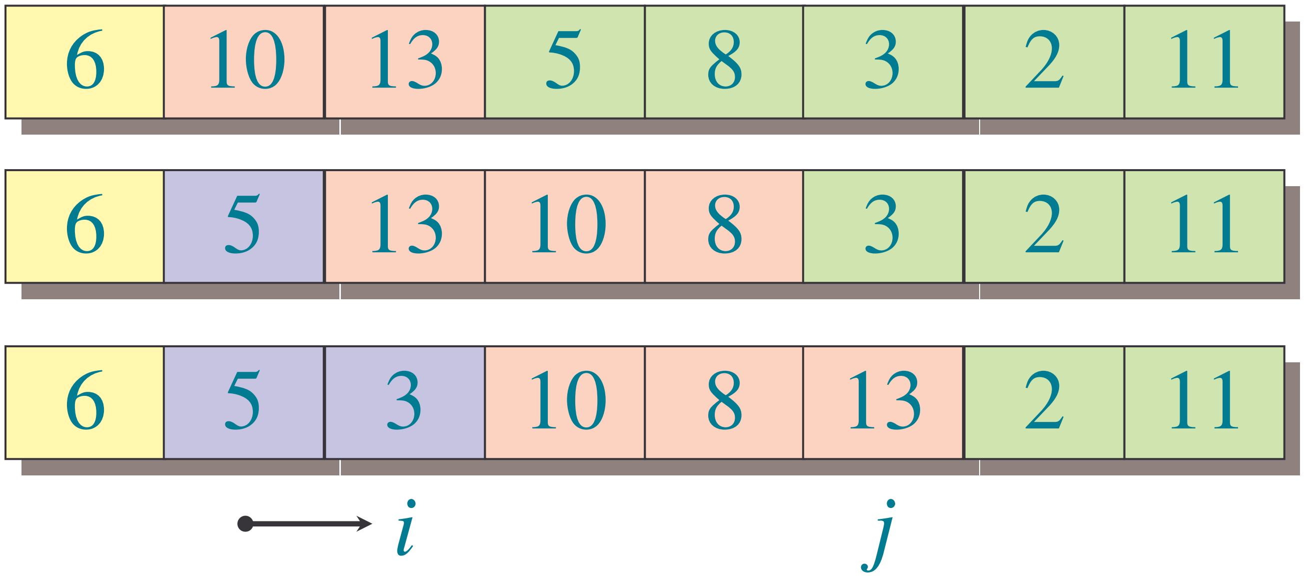


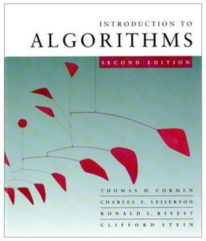
Example of partitioning



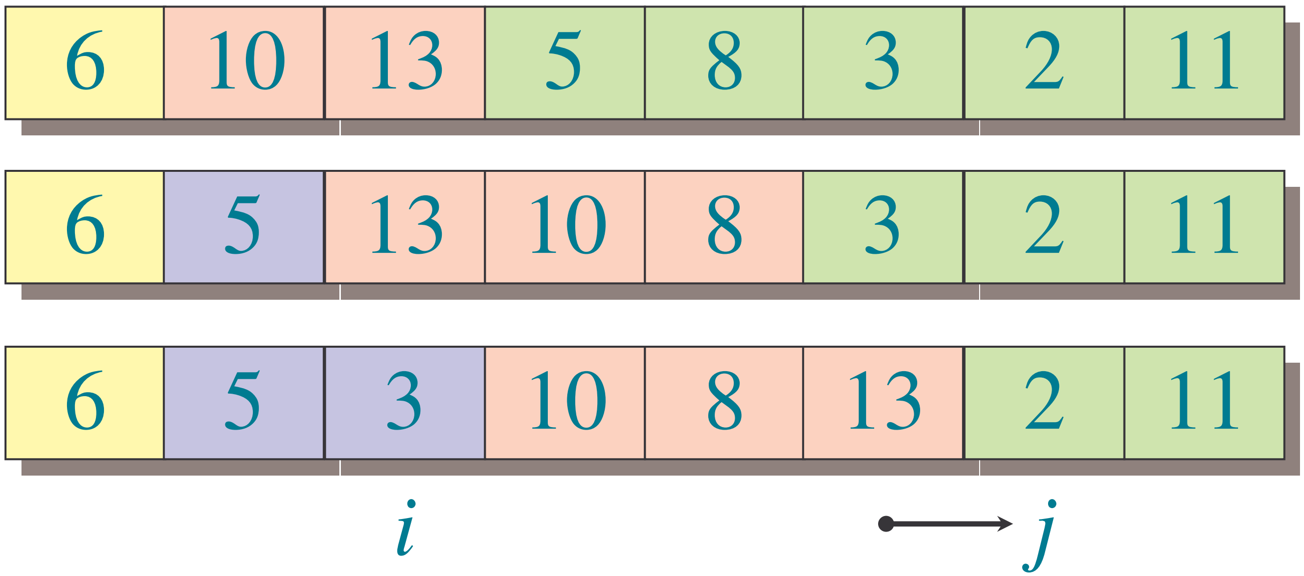


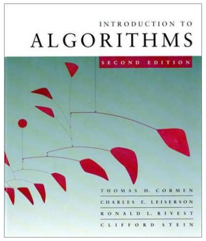
Example of partitioning



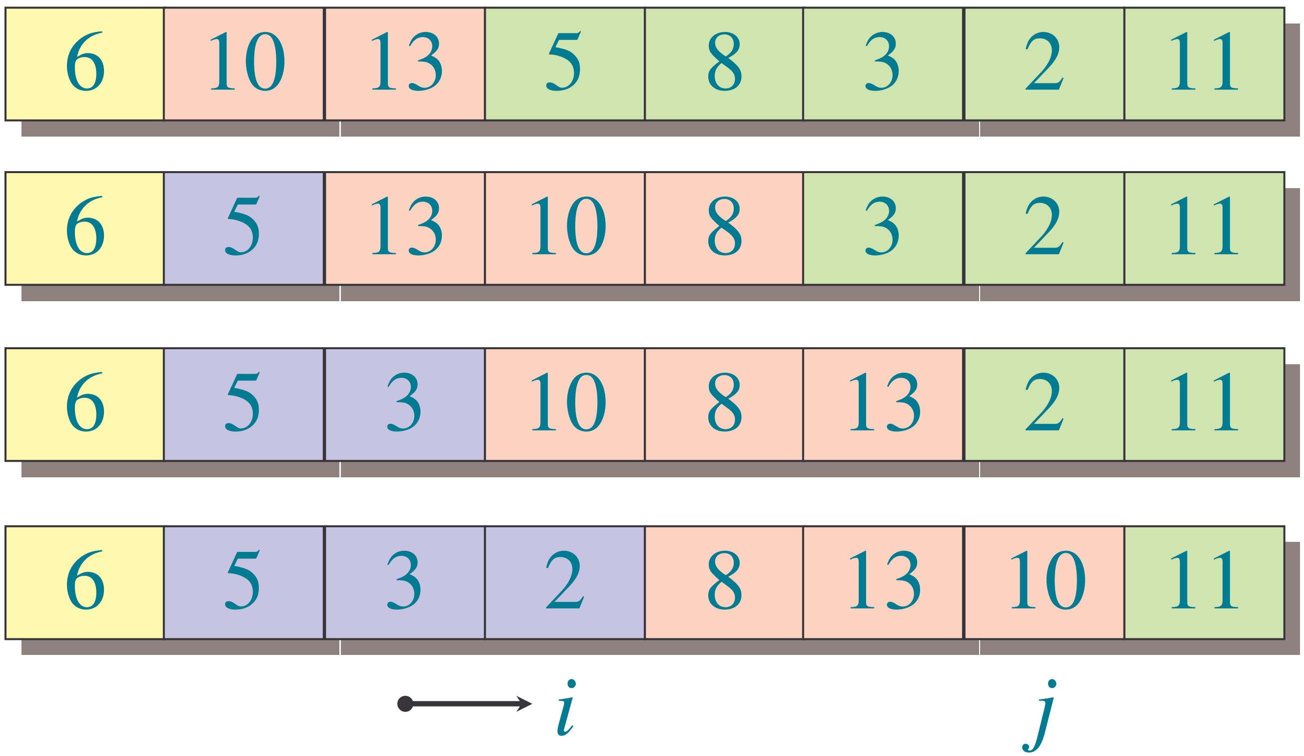


Example of partitioning

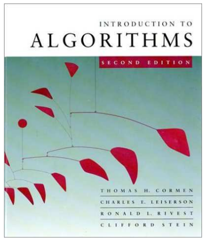




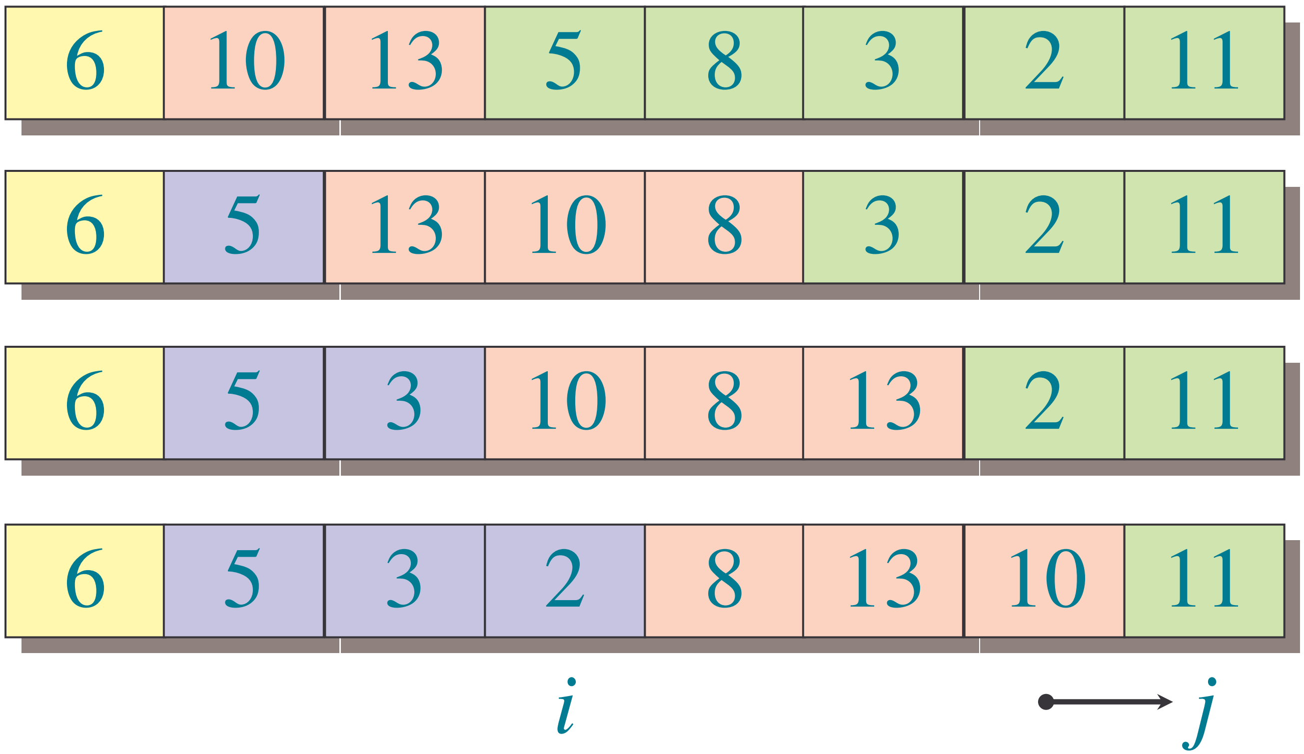
Example of partitioning

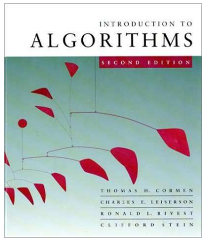


1/31/2007



Example of partitioning





Example of partitioning

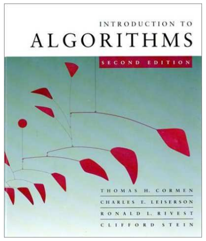
6	10	13	5	8	3	2	11

6	5	13	10	8	3	2	11

6	5	3	10	8	13	2	11

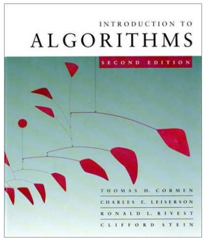
6	5	3	2	8	13	10	11

i $\qquad\qquad \longrightarrow j$



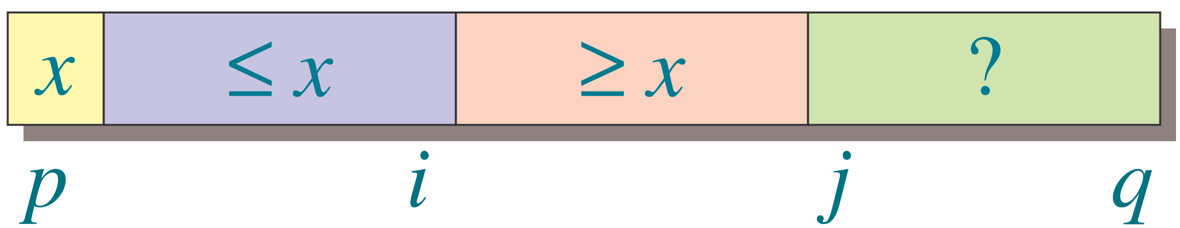
Example of partitioning

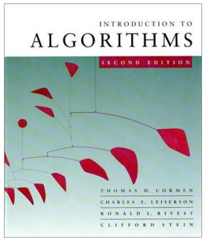
6	10	13	5	8	3	2	11
6	5	13	10	8	3	2	11
6	5	3	10	8	13	2	11
6	5	3	2	8	13	10	11
2	5	3	6	8	13	10	11
			i				



Invariant for the **for** loop

- Reminder: an *invariant* is a statement about a particular point in the code. The invariant should always be true when that point is reached (see CLRS Chapter 2).
- Goals of invariants: help *understand* and *explain* what the algorithm is doing, and help *prove* correctness.
- Example: In the PARTITION algorithm, before each execution of the **for** loop:
  $A[p] = x$,
  $A[p+1], \ldots, A[i] \leq x$, and
  $A[i+1], \ldots, A[j-1] \geq x$





Pseudocode for quicksort

QUICKSORT(A, p, r)

▷ Sort $A[p \,..\, q]$ in place.

if $p < r$

then $q \leftarrow$ PARTITION(A, p, r)

QUICKSORT$(A, p, q-1)$

QUICKSORT$(A, q+1, r)$

Initial call: QUICKSORT$(A, 1, n)$