جامعة طرابلس ـ كلية تقنية المعلومات

Design and Analysis Algorithms

تصميم و تحليل خوارزميات

ITGS301

Lecture 8 : المحاضرة الثامنة

Tripoli University - Faculty of Information Technology
Marwa Solla
m.solla@uot.edu.ly

Quick Sort

Quick sort is an other example of the ***divide-and-conquer*** approach, Proposed by C.A.R. Hoare in 1962, Sorts "in place" (like insertion sort, but not like merge sort).

- ***Divide:*** Partition the array into two sub arrays around a pivot x such that elements in lower sub array ≤ x ≤ elements in upper sub array.

- ***Conquer:*** Recursively sort the two sub arrays.
- ***Combine***: Finally, to put elements back into *array* in order,

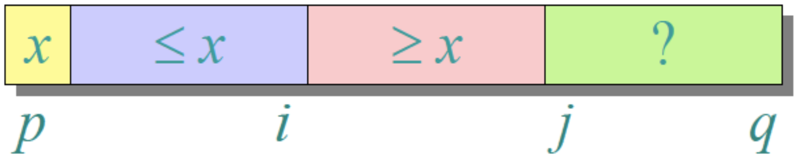
Basic idea:

1. Pick one element in the array, which will be the *pivot*.
2. Make one pass through the array, called a *partition* step, re-arranging the entries so that:
   - the pivot is in its proper place.

- entries smaller than the pivot are to the left of the pivot.
- entries larger than the pivot are to its right.

3. Recursively apply quicksort to the part of the array that is to the left of the pivot, and to the right part of the array.

Invariant:



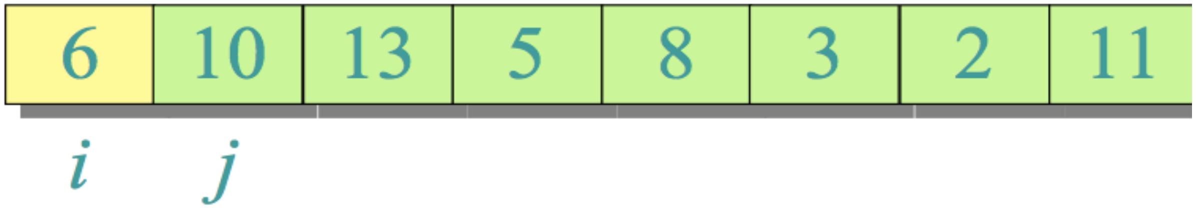
Here we don't have the merge step, at the end all the elements are in the proper order.

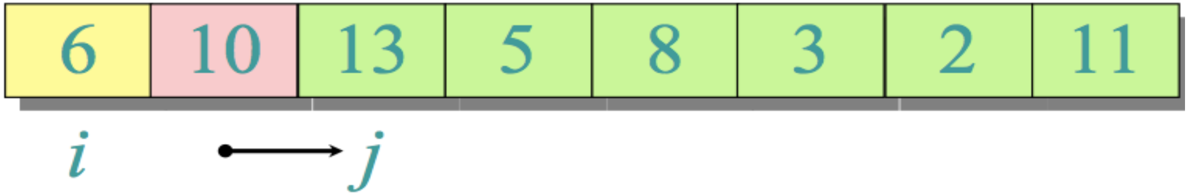
Quicksort algorithm:

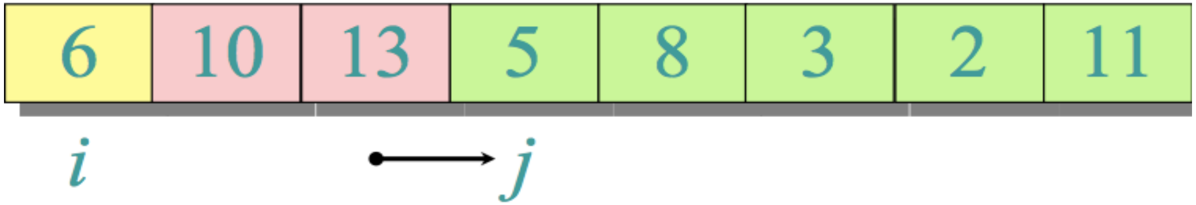
```
QUICKSORT(A, Left, Right)
 if Left < Right then
q ← PARTITION(A, Left, Right)
      QUICKSORT(A, left. q−1)
      QUICKSORT(A, q+1, right)
```

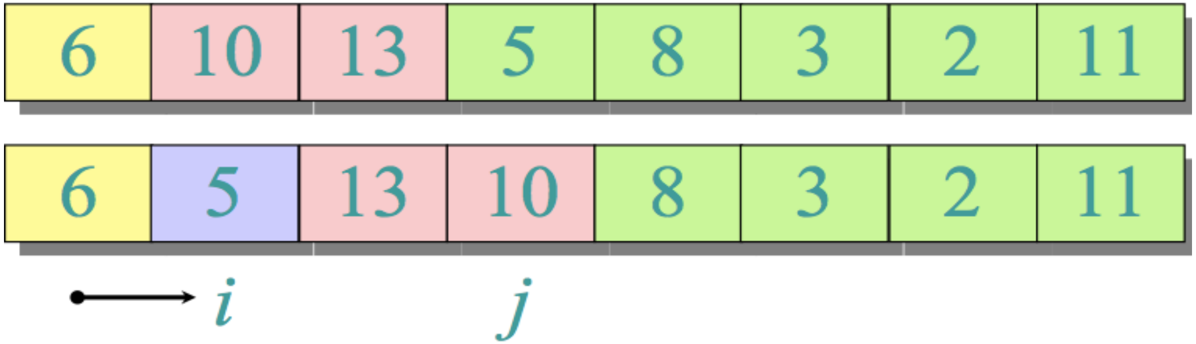
```
PARTITION(A, left, right)
      P = A[left]
       i= left
for (j = left+1; j <= right ; j++)
     if A[ j] <= P then
        i←i+1
        swap (A[i] , A[ j] )}
     end if
end for
swap(A[left], A[i])
return i
```

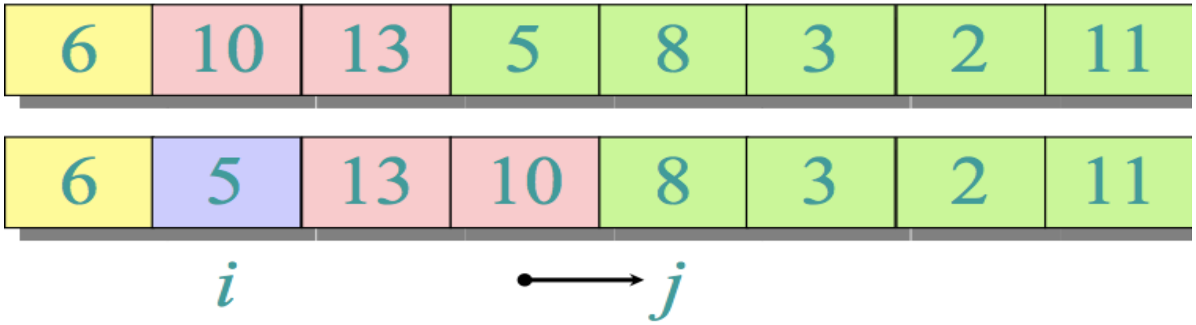
Example:

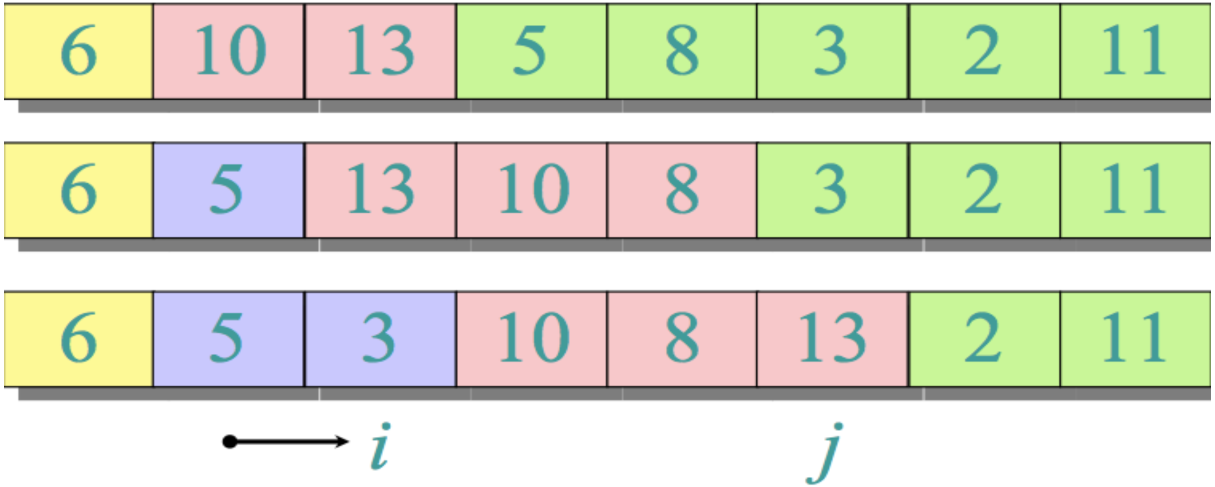


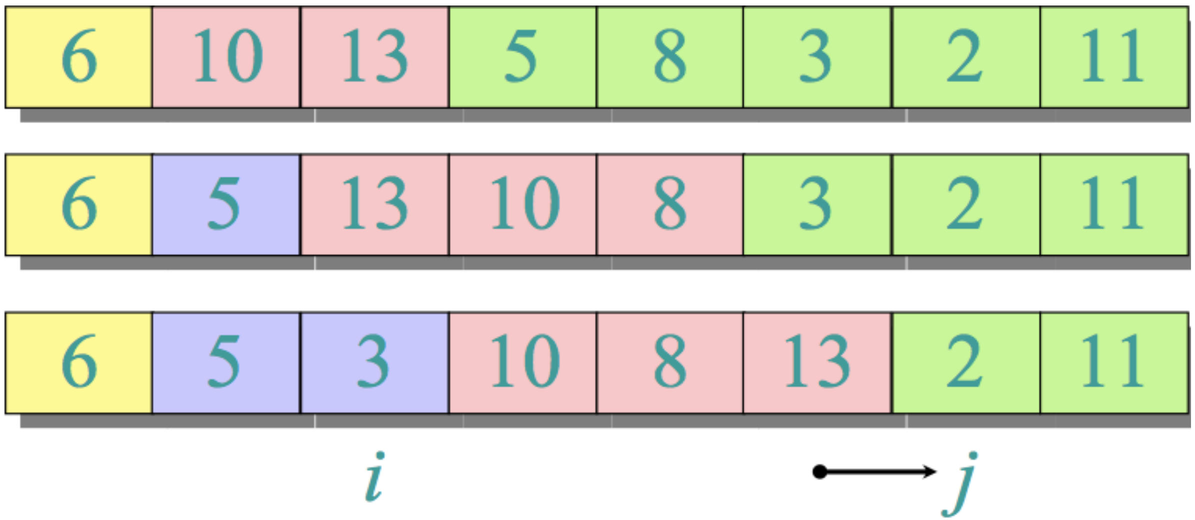

6
10
13
5
8
3
2
11
i
j

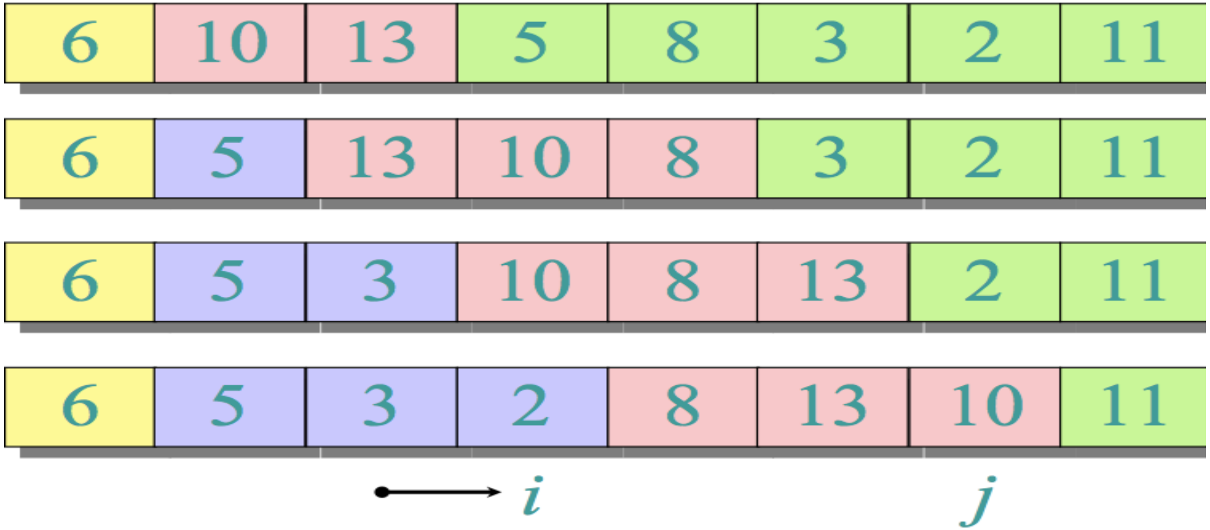

6
10
13
5
8
3
2
11
i
j

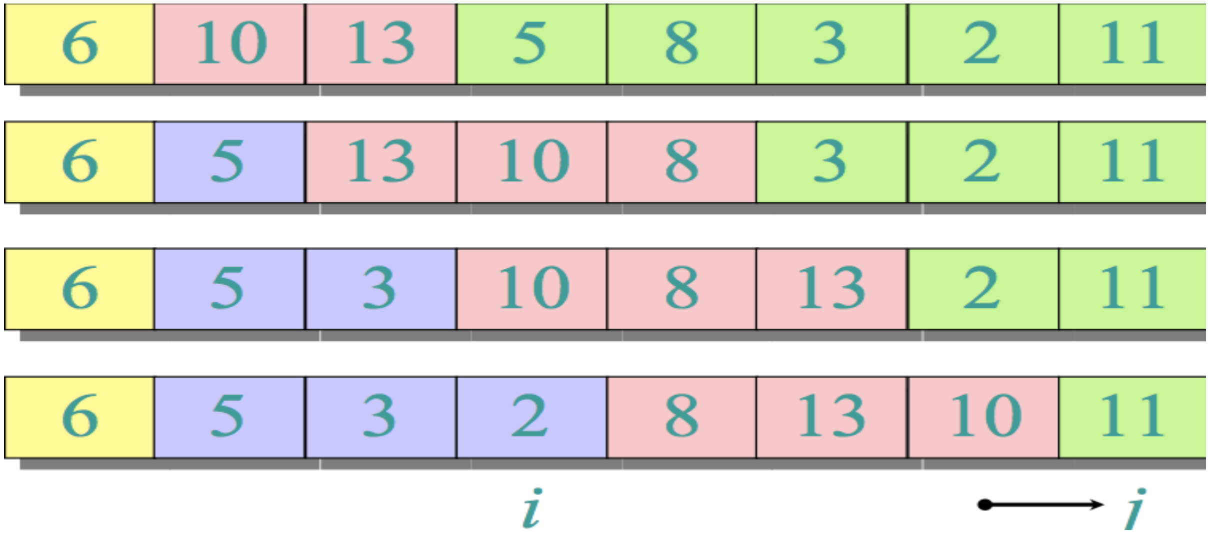

6 10 13 5 8 3 2 11
6 5 13 10 8 3 2 11
i
j

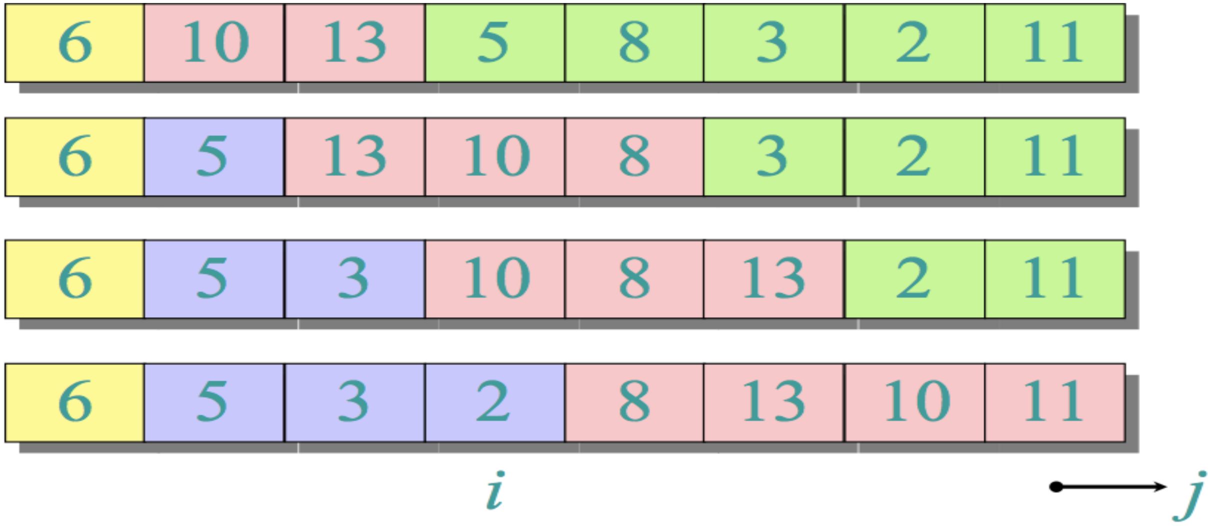

6
10
13
5
8
3
2
11
6
5
13
10
8
3
2
11
i
j

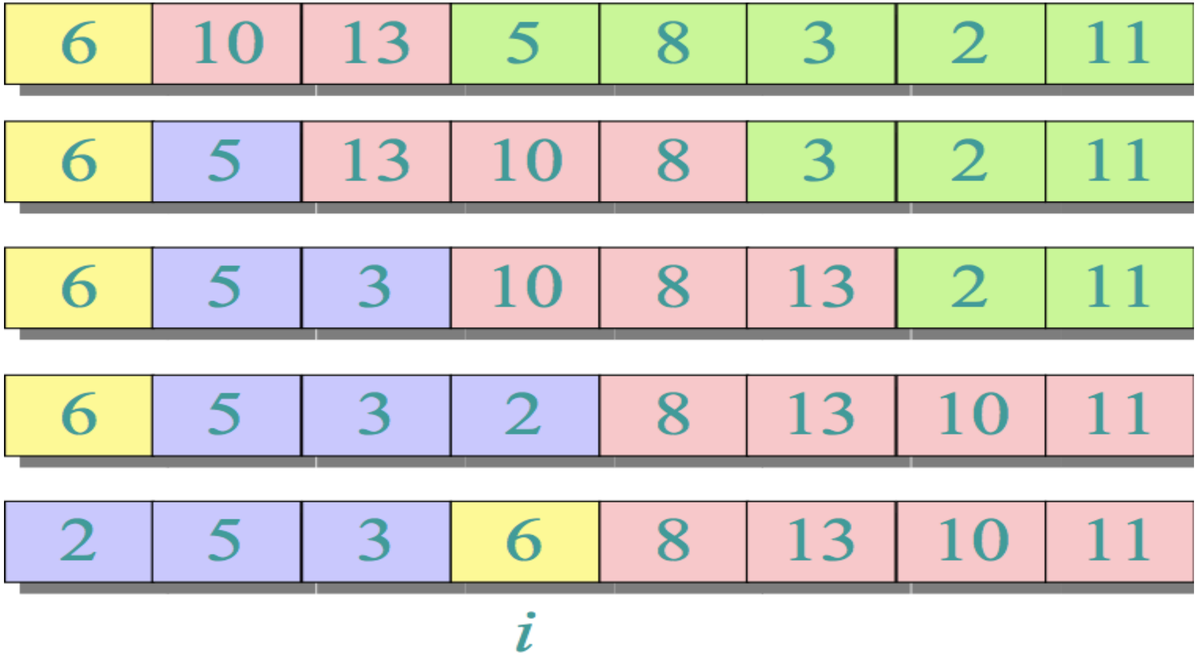

6 10 13 5 8 3 2 11
6 5 13 10 8 3 2 11
6 5 3 10 8 13 2 11
i
j


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


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
j


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
j


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
j


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

The running time for quick sort is described by following recurrence relation:

Worst case:

$$T(n) = \begin{cases} 1 & n=0 \\ T(n-1) + n & \text{for all } n > 0 \end{cases}$$

Best case:

$$T(n) = \begin{cases} 1 & n=1 \\ 2T(n/2) + n & \text{for all } n > 0 \end{cases}$$

Solve the following equation using iteration method

$$T(n) = \begin{cases} 1 & n=0 \\ T(n-1) + n & \text{for all } n > 0 \end{cases}$$

1 $T(n) = T(n-1) + n$

Since, $T(n-1) = T(n-1-1) + (n-1)$

$= T(n-2) + (n-1)$

2 then, $T(n) = T(n-2)+(n-1)+n$

$= T(n-2) + (n-1) + n$

Since, $T(n-2) = T(n-2-1) +(n-2)$

$= T(n-3) +(n-2)$

3 then, $T(n) = T(n-3) + (n-2) + (n-1) + n$

Since, $T(n-3) = T(n-3-1) + (n-3)$

$= T(n-4) + (n-3)$

4 then, $T(n) = T(n-4) + (n-3) + (n-2) + (n-1) + n$

.

.

Since, $T(n-k) = T(n-k+1) + (n-k)$

k $T(n) = T(n-k) + (n-(k-1)) + (n-(k-2)) \ldots + (n-1) + n$

Since $T(0) = 1$

$N-k = 0$ ⍰ $k=n$

$T(n) = T(n-n) + (n-(n-2)) + (n-(n-3)) \ldots + (n-1) + n$

$T(n) = T(n-n) + (n-n+1) + (n-n+2)(n-n+3) \ldots + (n-1) + n$

$T(n) = T(0) + 1 + 2 + 3 + 4 \ldots + (n-1) + n$

$T(n) = 1 + 1 + 2 + 3 + 4 \ldots + (n-1) + n$

$T(n) = 1 + n(n+1)/2$ ***$T(n) = O(n^2)$***

The End .