

## Chapter 4: Divide & Conquer

Quicksort (actually in chap 7)

sort  $A[p \dots r]$  recursively

Quicksort( $A, p, r$ )

1. if  $p < r$
2.  $q = \text{Partition}(A, p, r)$
3. Quicksort( $A, p, q-1$ )
4. Quicksort( $A, q+1, r$ )

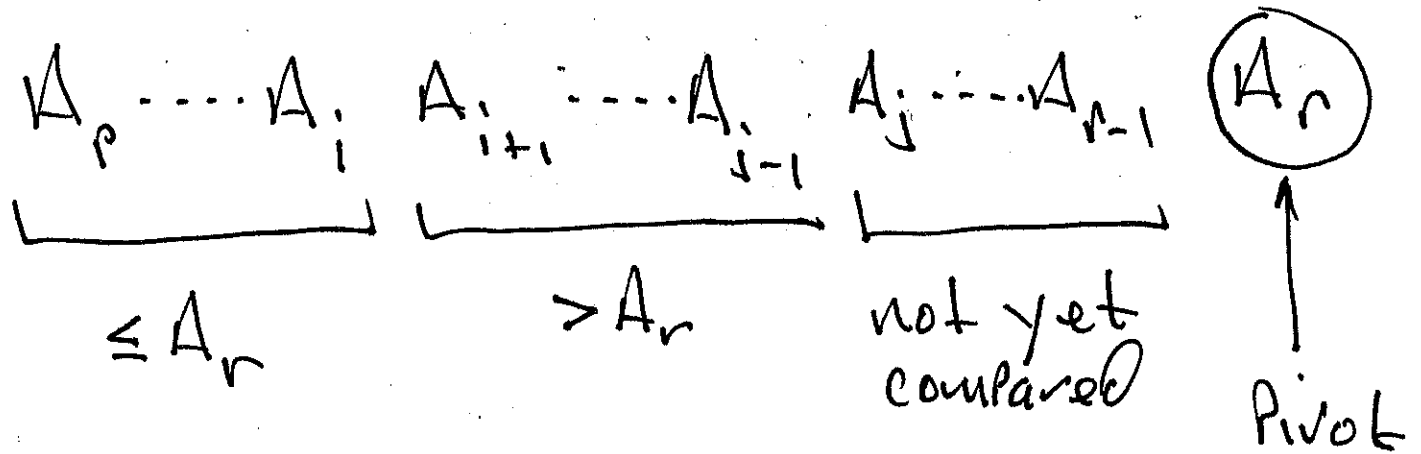
Partition cases:

$$A[p \dots (q-1)] \leq \underbrace{A[q]}_{\text{Pivot element}} < A[(q+1) \dots r]$$

Partition(A, p, q)

1.  $i = p - 1$
2. for  $j = p$  to  $(r-1)$
3.     if  $A[j] \leq A[r]$
4.          $i++$
5.          $A[i] \leftrightarrow A[j]$  (swap)
6.  $A[i+1] \leftrightarrow A[r]$  (swap)
7. return  $(i+1)$

What does partition do?



• if  $A_j \leq A_r$  :  $A_j \leftrightarrow A_{i+1}$  (swap  
 $i++$  and  $i++$

• if  $A_j > A_r$  :  $j++$

Runtime: count # of array comparisons.

$T(n)$  = worst case # of comparisons on arrays of length  $n$

At top level: Partition( $A, 1, n$ ) does  $n-1$  comparisons.

in worst case,  $A[1 \dots n]$  is already sorted.

$$T(n) = \begin{cases} 0 & n = 0, 1 \\ T(n-1) + (n-1) & n \geq 2 \end{cases}$$

# Iteration

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

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

$$\vdots$$

$$= \sum_{i=1}^k (n-i) + T(n-k)$$

stop when  $n-k=1$ , so  $k=n-1$

$$\therefore T(n) = \sum_{i=1}^{n-1} (n-i) + 0$$

$$= \sum_{i=1}^{n-1} n - \sum_{i=1}^{n-1} i$$

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

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