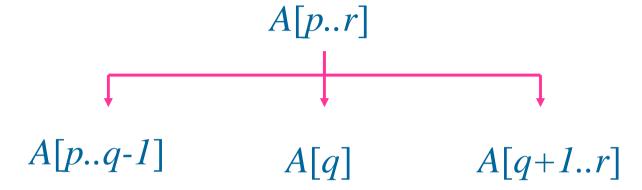
# **Quick Sort**

# Quick sort another divide-and-conquer algorithm

### Divide:



- ✓ Each element in  $A[p..q-1] \le A[q]$
- $\checkmark$  A[q]  $\le$  Each element in A[q+1..r]

# Conquer:

✓ Sort the two subarrays A[p..q-1] and A[q+1..r] by recursive calls to quick sort

#### Combine:

✓ Subarrays are sorted in place  $\Rightarrow$  *no combine step* 

Lecture 5

# Partitioning the Array

```
PARTATION( A, p, r)

1 - x \leftarrow A[r]

2 - i \leftarrow p - 1

3 - \text{for } j \leftarrow p \text{ to } r - 1

4 - \text{do if } A[j] \leq x

5 - \text{then } i \leftarrow i + 1

6 - \text{exchange } A[i] \leftrightarrow A[j]

7 - \text{exchange } A[i + 1] \leftrightarrow A[r]

8 - \text{return } i + 1
```

# Quick Sort Algorithm

```
QUICKSORT( A, p, r)

1- If p < r

2- then q \leftarrow \text{PARTATION}(A, p, r)

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

4- QUICKSORT (A, q+1, r)
```

# Performance of quick sort

#### Worst-Case partitioning

$$T(n) = T(n-1) + \Theta(n)$$
  $\Rightarrow$   $T(n) = \Theta(n^2)$ 

#### **Best-Case** partitioning

$$T(n) \le 2T(n/2) + \Theta(n) \implies T(n) = \Theta(n \lg n)$$

#### **Balanced-Case partitioning**

$$T(n) \le T(9n/10) + T(n/10) + \Theta(n)$$

$$\Rightarrow T(n) = \Theta(n \lg n)$$

#### Average-Case partitioning

Bad split  $\Rightarrow$  two subarrays of size 0 and n-1 Good split of (n-1) array  $\Rightarrow$  two subarray of size (n-1)/2 -1 and (n-1)/2

#### Resulting subarrays of the 2 splits

$$\Rightarrow 0, (n-1)/2 -1 \text{ and } (n-1)/2$$

$$T(n) = \Theta(n) + \Theta(n) = \Theta(n)$$
Lecture 5

## Randomized version of quicksort

#### RANDONIZED-PARTITION(A, p, r)

- 1-  $i \leftarrow \text{RANDOM}(p, r)$
- 2- exchange  $A[r] \leftrightarrow A[i]$
- 3- **return** PARTATION(A, p, r)

#### RANDOMIZED-QUICKSORT( A, p, r)

- 1- **If** p < r
- 2- then  $q \leftarrow \text{RANDOMIZED-PARTATION}(A, p, r)$
- 3- RANDOMIZED-QUICKSORT (A, p, q-1)
- 4- RANDOMIZED-QUICKSORT ( A, q+1, r)

# Background for Analyzing the Running Time of Randomized Quick Sort

#### Indicator Random Variable

For a given space S and event A, the *indicator random variable*  $I\{A\}$  is defined as:

$$I\{A\} = \begin{cases} 1 & if & A & occurs, \\ 0 & if & A & does & not & occurs. \end{cases}$$

For a space  $S = \{M,N\}$  each with probability  $\frac{1}{2}$ The indicator random variable:

$$\mathbf{X}_{M} = \mathbf{I}\{Y = M\} = \begin{cases} 1 & if \quad Y = M, \\ 0 & if \quad Y = N. \end{cases}$$

#### **Expectation**

$$E[X_M] = E[I{Y=M}]$$
  
= 1 • Pr{Y=M} + 0 • Pr{Y=N}  
= 1 • (1/2) + 0 • (1/2)  
= 1/2

$$E[X_A] = Pr\{A\}$$

#### Running Time of Randomized Quick Sort

$$ightharpoonup T(n) = O(n + X)$$

X # number of comparison over entry execution of Quick-sort on an element array.

$$X = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} X_{ij}$$

- $\triangleright$  E[X]= $O(n \lg n)$
- $ightharpoonup T(n) = O(n \lg n)$