Quick Sort

Quick sort

↓

another divide-and-conquer algorithm

Divide:

$A[p..r]$

$A[p..q\text{-}1]$ $\quad$ $A[q]$ $\quad$ $A[q+1..r]$

- ✓Each element in $A[p..q\text{-}1] \leq A[q]$
- ✓$A[q] \leq$ Each element in $A[q+1..r]$

Conquer:

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

Combine:

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

Partitioning the Array

```
PARTATION( A, p, r)
1- x ← A[r]
2- i ← p - 1
3- for j ← p to r-1
4-         do if A[j] ≤ x
5-               then i ← i +1
6-                   exchange A[i] ↔ A[j]
7- exchange A[i+1] ↔ A[r]
8- return i+1
```

Quick Sort Algorithm

QUICKSORT(A, *p*, *r*)

1- **If** $p < r$

2- **then** $q \leftarrow$ PARTATION (A, *p*, *r*)

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

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

Performance of quick sort

Worst-Case partitioning

$$T(n) = T(n\text{-}1) + \Theta(n) \quad \Rightarrow \quad T(n) = \Theta(n^2)$$

Best-Case partitioning

$$T(n) \leq 2T(n/2) + \Theta(n) \quad \Rightarrow \quad T(n) = \Theta(n \lg n)$$

Balanced-Case partitioning

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

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

Average-Case partitioning

<u>Bad split</u> $\Rightarrow$ two subarrays of size 0 and n-1

<u>Good split</u> of (n-1) array $\Rightarrow$ two subarray of size (n-1)/2 -1 and (n-1)/2

<u>Resulting subarrays of the 2 splits</u>

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

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

Randomized version of quicksort

RANDONIZED-PARTITION(A, *p*, *r*)
1- $i \leftarrow$ 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** $q \leftarrow$ 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:

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

For a space S = {M,N} each with probability ½

The indicator random variable:

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

Expectation

$E[X_M] = E[I\{Y{=}M\}]$

$= 1 \bullet \Pr\{Y{=}M\} + 0 \bullet \Pr\{Y{=}N\}$

$= 1 \bullet (1/2) + 0 \bullet (1/2)$

$= 1/2$

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

Running Time of Randomized Quick Sort

- T(n) = O(n + X)

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

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

- E[X]=O(n lg n)

- T(n) =O(n lg n)