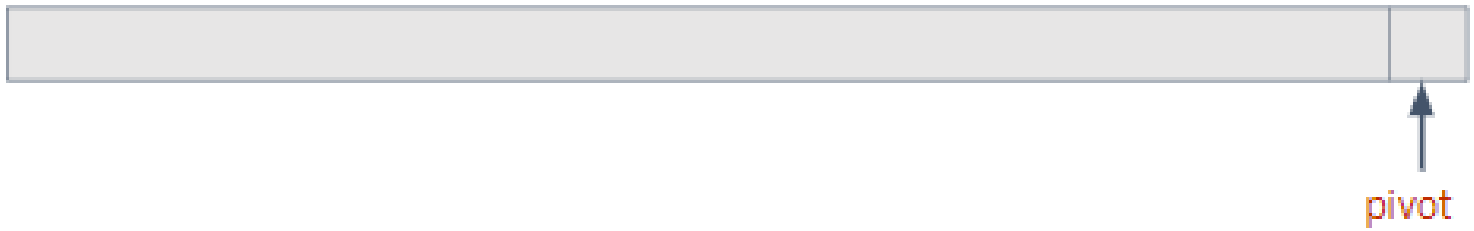
MCAC 302: Design and Analysis of Algorithms

Neelima Gupta

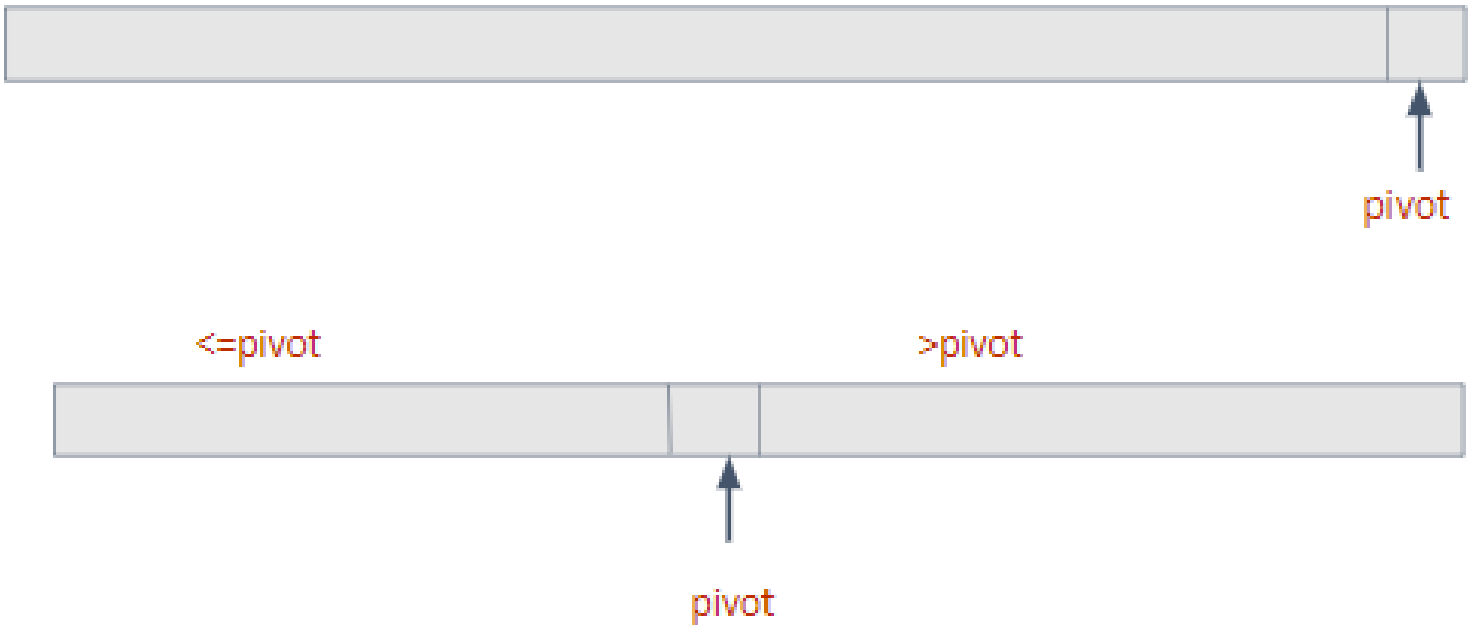
ngupta@cs.du.ac.in

September 3, 2020

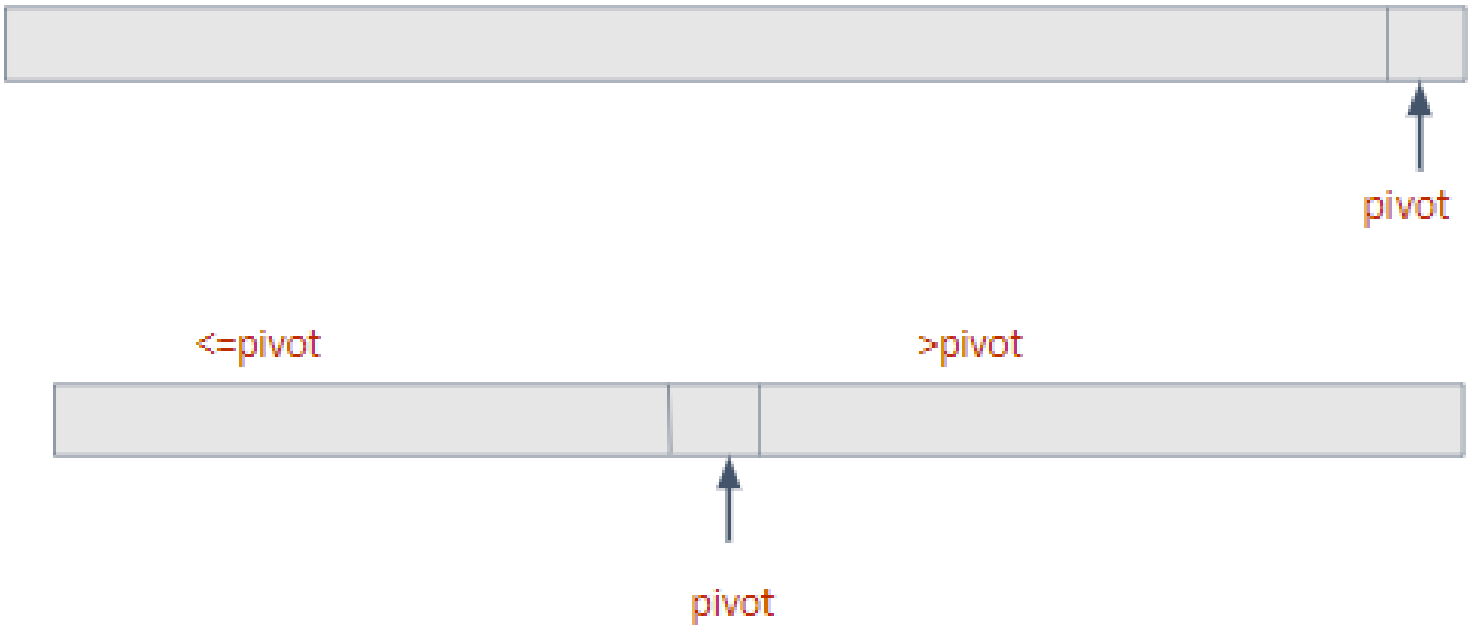
Quick Sort



Quick Sort

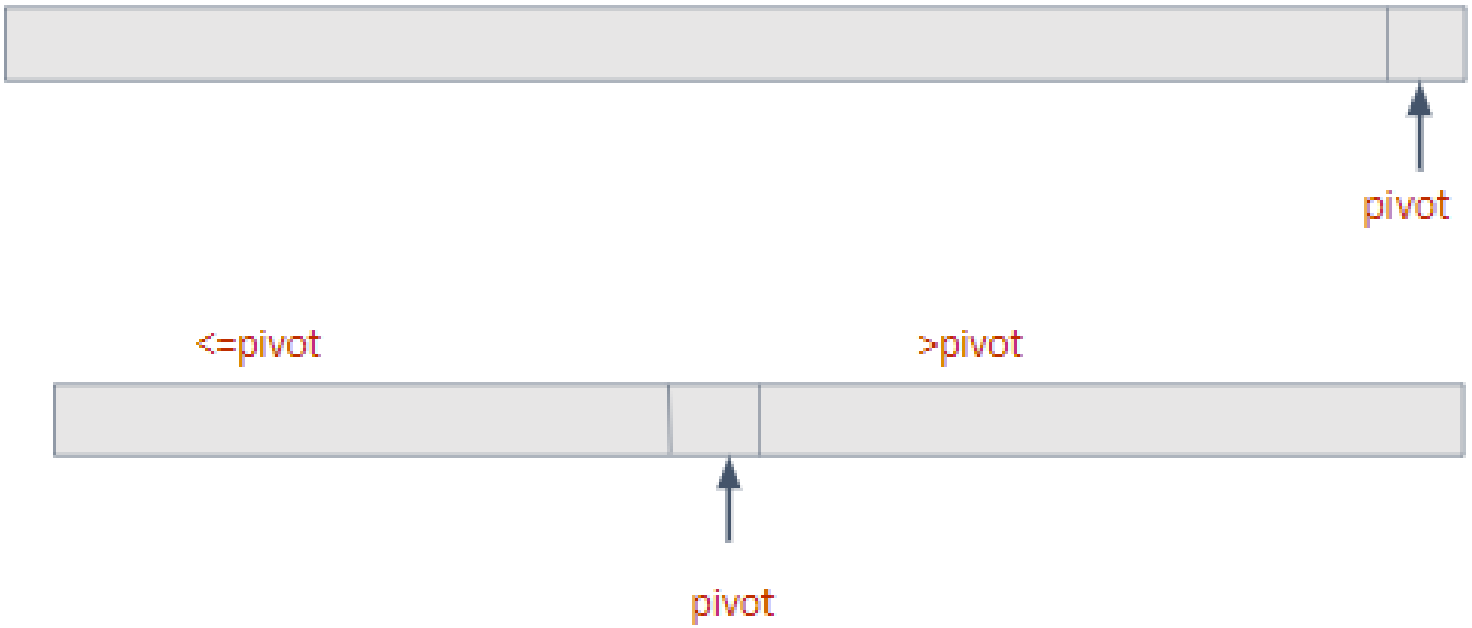


Quick Sort



10, 25, 8, 9, 21, 15, 10 $pivot = 10$

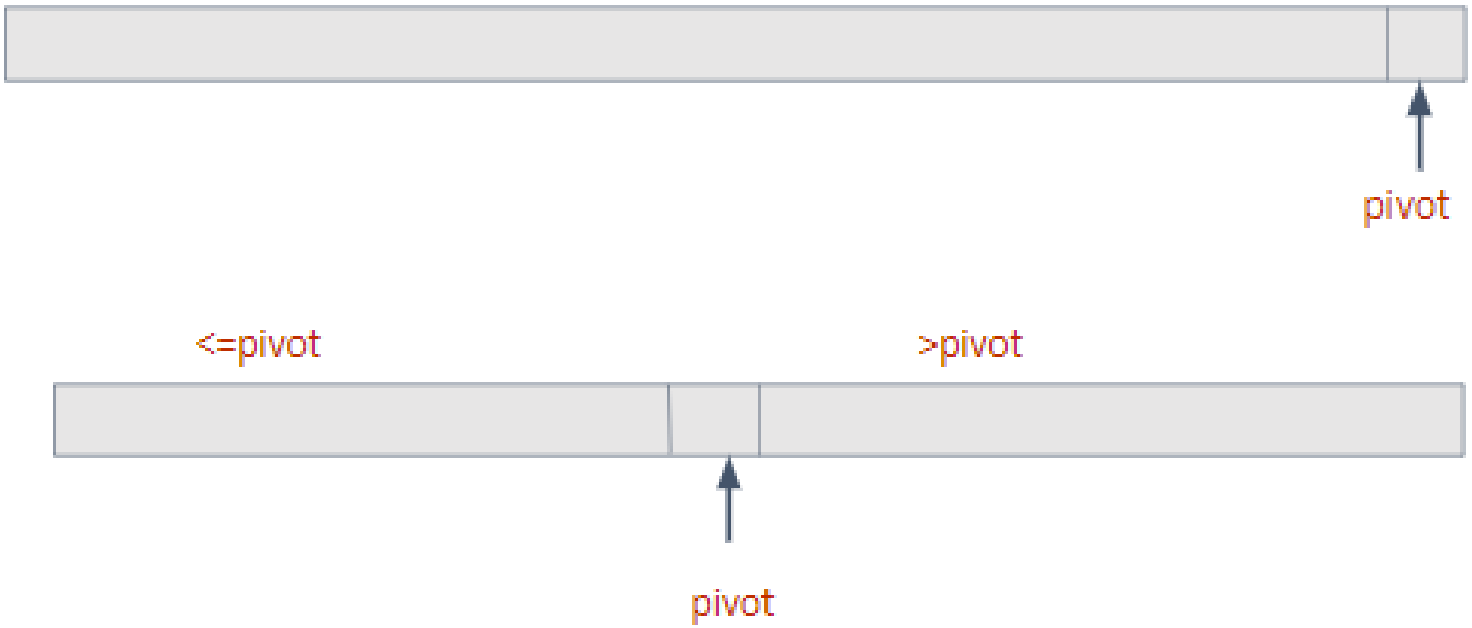
Quick Sort



10, 25, 8, 9, 21, 15, 10 $\quad$ $pivot = 10$

10, 8, 9, 10, 25, 21, 15 $\quad$ (after partition)

Quick Sort



10, 25, 8, 9, 21, 15, 10 $\quad$ *pivot* = 10

10, 8, 9, 10, 25, 21, 15 $\quad$ (after partition)

8, 9, 10, 10, 15, 21, 25

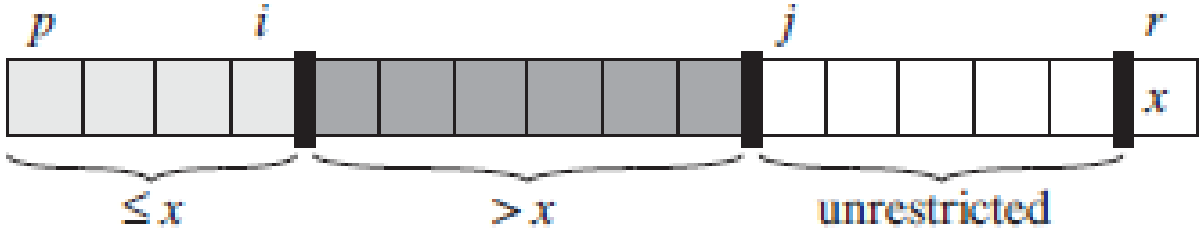
(after sorting the partitions recursively)

Quick Sort

input : Array: $p, r, A[p \ldots r]$
output: Sorted Array: $A[p] \leq A[p+1] \leq .... \leq A[r]$

```
QuickSort(A,p,r)
/* Performs sorting on the input array */
if p < r then
    q=Partition(A,p,r)
    QuickSort(A,p,q-1)
    QuickSort(A,q+1,r)
end
```

Partition



[1]

$pivot = x$

Invariance:

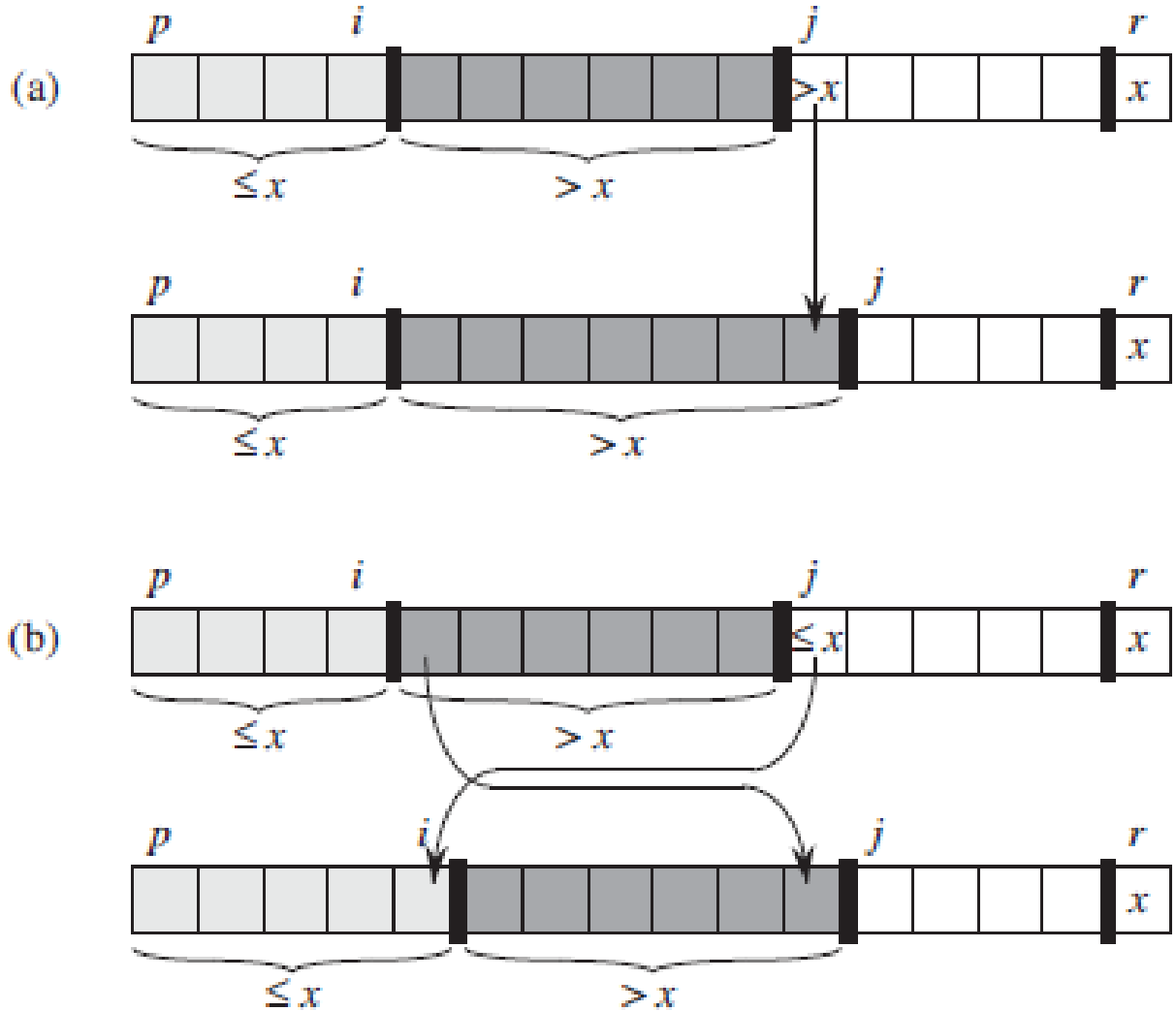
- $A[p \ldots i] \leq pivot$
- $A[i+1 \ldots j-1] > pivot$
- $A[r] = pivot$

Initially, if $i = p - 1$ and $j = p$, first two invariance properties are satisfied vacuously.

[1] Figure from Cormen, Leiserson, Rivest, Stein: Introduction to Algorithms

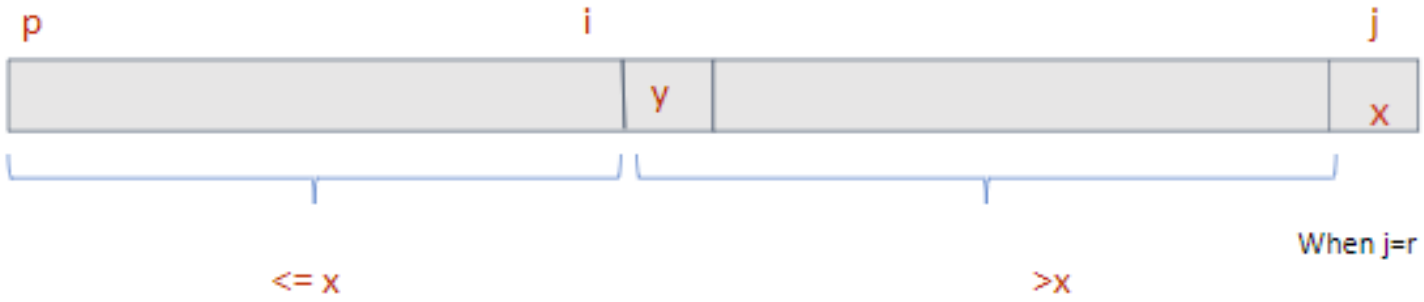
Partition contd..

Invariance: $A[p \ldots i] \leq pivot$, $A[i+1 \ldots j-1] > pivot$, $A[r] = pivot$

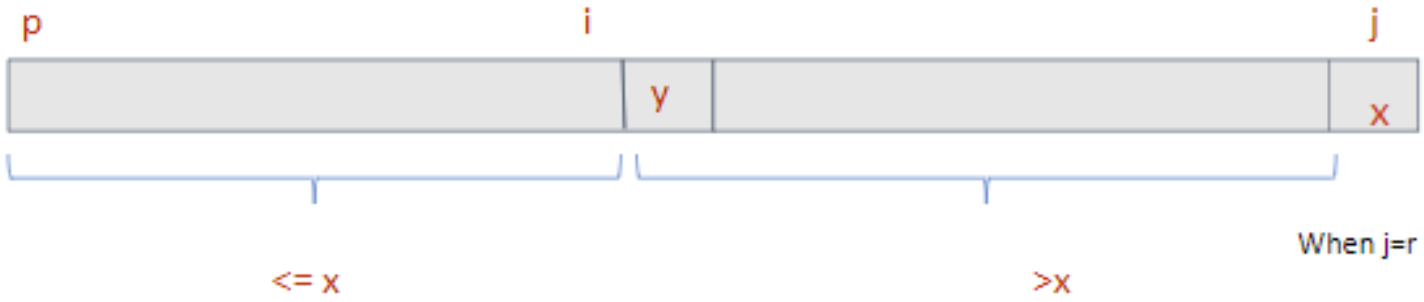


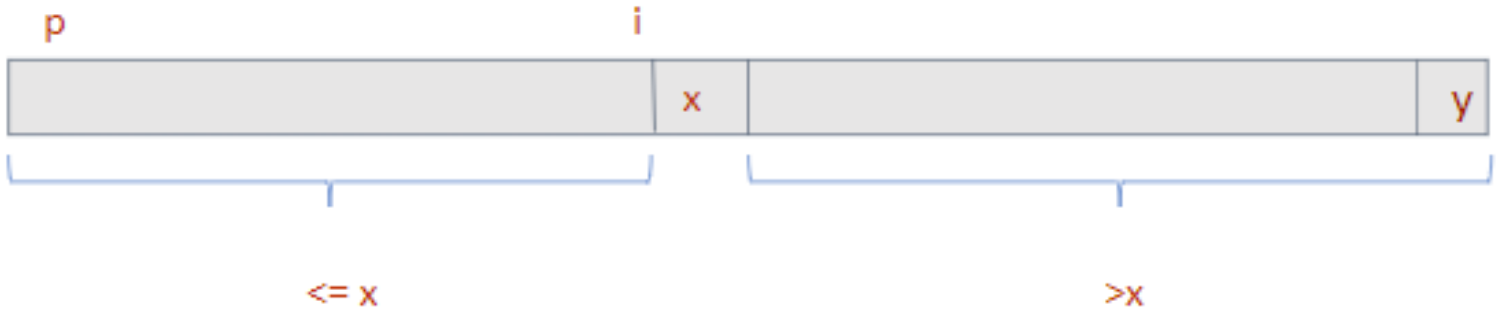
[2]Figure from Cormen, Leiserson, Rivest, Stein: Introduction to Algorithms

Partition contd..



Partition contd..





Partition Pseudo-code

input : Array: $p, r, A[p \ldots r]$
output: q: the Index of the pivot

```
Partition(A, p, r)
/* "p" and "r" are the first and the last indices, respectively,
  of the array A */
x=A[r]
i=p-1
for j : p to r − 1 do
    if A[j] ≤ x then
        i=i+1
        exchange A[i] with A[j]
    end
end
exchange A[i+1] with A[r]
return i+1
```