Example: Randomized Quick Sort

Problem: The PARTITION algorithm for "standard" quick sort uses the first element as the pivot element, which may in practice not be a "typical" element (e.g., consider the case that the data is already sorted). Also, although it works for duplicates, performance can degrade because all equal elements will end up on the same side and may cause the recursion to be unbalanced.

Idea: Use a slightly more complex randomized "3-partitioning" procedure. Instead of choosing the pivot element to be $A[i]$, choose a random index $i \leq r \leq j$ and exchange $A[r]$ with $A[i]$. A third index variable $z$ allows for $<, =,$ and $>$ regions:
(randomized quick sort continued)

**function 3PARTITION**(*i*,*j*)

\[ r := \text{a random integer in the range } i \leq r \leq j \]

Exchange *A*[*i*] and *A*[*r*].

\[ x := i+1 \]

\[ y := z := j \]

while \( x \leq y \) do

while \( x \leq y \) and *A*[*x*] < *A*[*i*] do \( x := x+1 \)

while \( x \leq y \) and *A*[*y*] ≥ *A*[*i*] do begin

if *A*[*y*] = *A*[*i*] then \( y := y-1 \)

else if *A*[*y*] > *A*[*i*] then begin

exchange *A*[*y*] and *A*[*z*] 

\[ y := y-1 \]

\[ z := z-1 \]

end

end

if \( x < y \) then begin

exchange *A*[*x*] and *A*[*y*] 

\[ x := x+1 \]

\[ y := y-1 \]

end

end

Exchange *A*[*i*] and *A*[*y*].

**return** \( y-1 \) and \( z+1 \) (that is, *A*[*i*]...*A*[*y*−1] < *A*[*y*]...*A*[*z*] < *A*[*z*+1]...*A*[*j*])

end
(randomized quick sort continued)

**Complexity:** Since the time for 3PARTITION differs by a constant factor from PARTITION, the expected time for quick sort remains $O(n \log(n))$.

** However, now this bound is independent of the input distribution, even when duplicates are allowed.

**An alternate approach:** Use a random pivot element and modify PARTITION so that the equal elements are distributed equally between the left and right. We no longer have the benefit of smaller sub-problems when there are many equal elements, but on the other hand, we do not have the overhead of 3-partitioning when there are not many equal elements.