Quicksort

CLRS Reading: Sections 7.1, 7.2, pages 170 – 179
Partitioning

Partition(A, p, r)

\[ x = A[r] \quad // \text{serves as pivot} \]
\[ i = p - 1 \]

for \( j = p \) to \( r - 1 \)

if \( A[j] \leq x \)

\[ i = i + 1 \]

exchange \( A[i] \) with \( A[j] \)

exchange \( A[i + 1] \) with \( A[r] \)

return \( i + 1 \)

At the beginning of each iteration,

1. If \( p \leq k \leq i \), then \( A[k] \leq x \).
2. If \( i + 1 \leq k \leq j - 1 \), then \( A[k] > x \).
3. If \( k = r \), then \( A[k] = x \).

*Remember, we must show initialization; maintenance; and termination.
Maintaining Loop Invariant

Partition($A$, $p$, $r$)
$\begin{align*}
  x &= A[r] \\
  i &= p-1 \\
  \text{for } j = p \text{ to } r-1 \\
  \text{if } A[j] \leq x & \rightarrow i = i+1 \\
  \text{exchange } A[i] \text{ with } A[j] \\
  \text{exchange } A[i+1] \text{ with } A[r] \\
  \text{return } i+1
\end{align*}$

Running Time of Partition

Partition($A$, $p$, $r$)
$\begin{align*}
  x &= A[r] \quad \text{// serves as pivot} \\
  i &= p-1 \\
  \text{for } j = p \text{ to } r-1 \\
  \text{if } A[j] \leq x & \rightarrow i = i+1 \\
  \text{exchange } A[i] \text{ with } A[j] \\
  \text{exchange } A[i+1] \text{ with } A[r] \\
  \text{return } i+1
\end{align*}$
Correctness of Quicksort*

Quicksort(A, p, r)
if p < r
    q = Partition(A, p, r)
    Quicksort(A, p, q-1)
    Quicksort(A, q+1, r)

*By induction on the size of the subproblem, using correctness of Partition algorithm.

The Best

- Intuitively, the best we can hope for is a balanced partition.
The Worst

Intuitively, the worst we can imagine is a totally unbalanced partition:

*Could such a thing actually happen?
T(n) = T(n-1) + n

Randomized-Partition(A, p, r)
    i = Random(p, r)
    exchange A[r] with A[i]
    return Partition(A, p, r)

Randomized-Quicksort(A, p, r)
    if p < r
        q = Randomized-Partition(A, p, r)
        Randomized-Quicksort(A, p, q-1)
        Randomized-Quicksort(A, q+1, r)

Making the Worst Case Less Likely