Data Structures and Algorithms

Lecture 08

Aniket Basu Roy

2026-01-30 Fri

Contents

1 Agenda

1.1 Quicksort

- Recap
- Proof of Correctness
- Time Complexity

1.2 An Example

- 8 1 6 4 0 3 9 5

1.3 Loop Invariant

1.3.1 Define

- Let $x = A[r]$
- $LEFT := A[p : i]$
- $RIGHT : A[i + 1 : j - 1]$
- $UNKNOWN : A[j : r - 1]$

1.3.2 Observe

- i is th last index of $LEFT$
- j is the first index of $UNKNOWN$

1.3.3 Loop Invariant

- $\forall a \in LEFT,\ a \leq x$
- $\forall b \in RIGHT,\ b > x$

1.4 Draw the flow digraph

- Initialization
  - $LEFT : A[p : p - 1] = \emptyset$
  - $RIGHT : A[p : p - 1] = \emptyset$
- Maintenance
  - If-statement is true ($A[j] \leq x$)
    * $i + +$
    * $A[i] \leftrightarrow A[j]$
    * $j + +$
    * $\forall a \in LEFT,\ a \leq x$
    * $\forall b \in RIGHT,\ b > x$
  - Else ($A[j] > x$)
    * $j + +$
    * $\forall a \in LEFT,\ a \leq x$
    * $\forall b \in RIGHT,\ b > x$

- Termination
  - $j = r$
  - $A[j : r-1] = \emptyset$

1.5 Worst-case time complexity

$$T(n) = \max_{0 \le q \le n-1} \big\{T(q) + T(n-q-1)\big\} + \Theta(n)$$

- Guess: $T(n) \le cn^2$
- $T(n) \le \max_{0 \le q \le n-1} \big\{cq^2 + c(n-q-1)\big\} + \Theta(n)$
- $c \max_{0 \le q \le n-1} \big\{q^2 + q^2 + (n-1)^2 - 2q(n-1)\big\} + \Theta(n)$
- $c \max_{0 \le q \le n-1} \big\{2q(q-(n-1)) + (n-1)^2\big\} + \Theta(n)$
  - As $q \le n-1,\ 2q(q-(n-1)) \le 0$
    * $\le (n-1)^2$

Plugging $(n-1)^2$

- $T(n) \le c(n-1)^2 + \Theta(n)$
- $\le cn^2 - c(2n-1) + \Theta(n)$

  Choose c appropriately, such that $c(2n-1)$ dominates $\Theta(n)$.

- $T(n) \le cn^2$

2 Comparison-Based Sorting

- What is a comparison-based sorting algorithm?
- Lower Bounds using Decision Trees (full binary tree, viz., both children present or a leaf).
- Internal Nodes represent the comparisons.
- A Leaf represents a sorted array. For every sorted sequence there exists a leaf.
- A directed path from the root to a leaf represents the sequence of comparisons made by the algorithm comparison-based algorithm.

- If there are ℓ leaves, then $n! \leq \ell$.
- If h is the height of the tree, then $\ell \leq 2^h$.
- Using Stirling's approximation, $h = \Omega(n \log n)$.