Data Structures and Algorithms

Lecture 07

Aniket Basu Roy

2026-01-27 Wed

Contents

1 Agenda

1.1 Quicksort

- Description
- Proof of Correctness
- Time Complexity

1.2 Divide

Pick a pivot element and find its final position in the sorted array A[q]

1.3 Conquer

The final position of the pivot element partitions the array into two parts. Run the Quicksort algorithm recursively on the two parts.

1.4 Combine

1.5 An Example

- 8 1 6 4 0 3 9 5

1.6 Proof of Correctness

- Induction on the number of array elements
- Loop invariant inside the PARTITION function

1.7 Time Complexity

1.7.1 Balanced Partitions

- Worst-case partitions
  - 0 and $n - 1$
  - Ex: sorted array
- Best-case partitions
  - .5 and .5
  - Ex:
- Balanced partitions
  - with a constant ration .1 and .9

The recurrence relations for each type of the partition.

1.8 Worst-case time complexity

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

- Guess: $T(n) \leq cn^2$