Data Structures and Algorithms

Lecture 02

Aniket Basu Roy

BITS Pilani Goa Campus

2026-01-12 Mon

Agenda

Insertion Sort

Proof of Correctness

Counting Steps

Problem: Sorting

Input: A collection of numbers, $a_1, a_2, \ldots, a_n$

Output: Sorted sequence, $a_{i_1} \leq a_{i_2} \leq \cdots \leq a_{i_n}$

Insertion Sort

```
INSERTION-SORT(A, n)
1  for i = 2 to n
2      key = A[i]
3      // Insert A[i] into the sorted subarray A[1 : i − 1].
4      j = i − 1
5      while j > 0 and A[j] > key
6          A[j + 1] = A[j]
7          j = j − 1
8      A[j + 1] = key
```

Insertion Sort

Proof of Correctness

Loop invariant: The prefix of the array is sorted

- Initialization: A[1] is sorted
- Maintenance: If A[1:j-1] is sorted then A[1:j] is sorted
- Termination: A[1:n] is sorted

Insertion Sort

Counting Steps

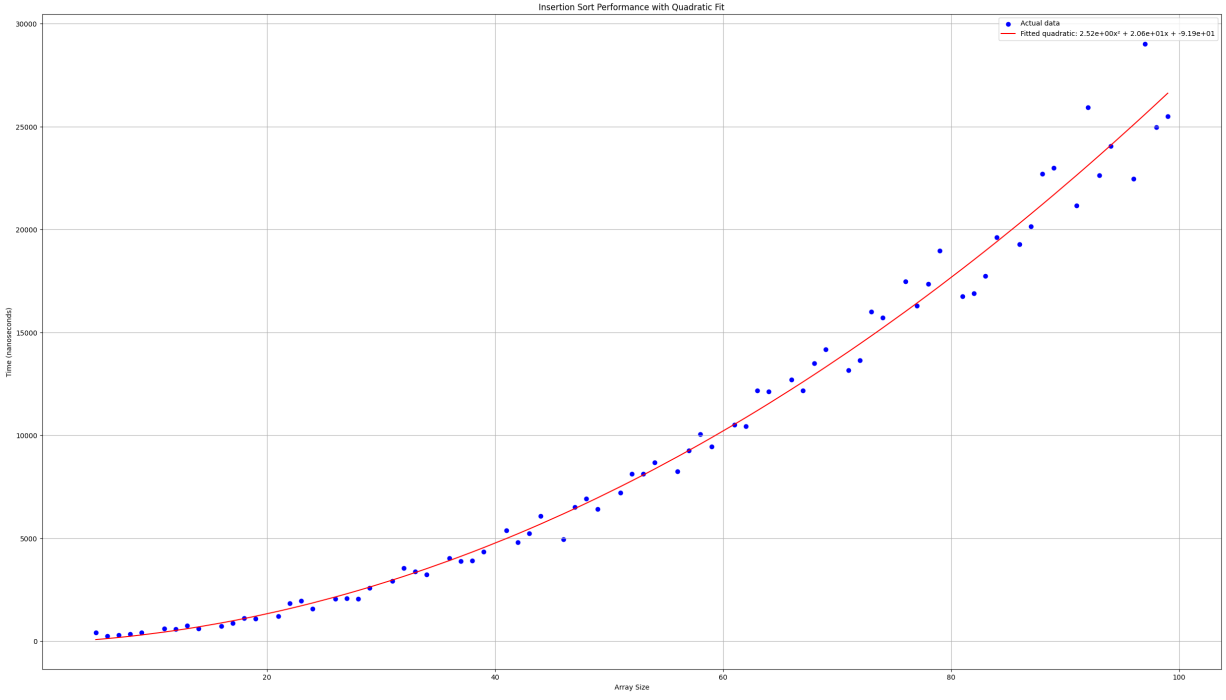
Insertion Sort

Counting Steps

- Best-Case Analysis
- Average-Case Analysis
- Worst-Case Analysis

Insertion Sort

Backed by Data



Order of Growth

Asymptotic Notation

- O-notation
- Ω-notation
- Θ-notation