This chapter introduces the fundamental role of algorithms in computing, emphasizing their importance as a technology and their impact on system performance.

---

1. The Role of Algorithms in Computing

1.1 Algorithms

Informally, an **algorithm** is any well-defined computational procedure that takes some value (or set of values) as input and produces some value (or set of values) as output in a finite amount of time. It is a sequence of computational steps that transform the input into the output, serving as a tool for solving well-specified computational problems.

The Sorting Problem

The sorting problem is a fundamental operation in computer science often used as an intermediate step in other programs.

- **Input:** A sequence of n numbers $\langle a_1, a_2, \ldots, a_n \rangle$.
- **Output:** A permutation (reordering) $\langle a'_1, a'_2, \ldots, a'_n \rangle$ of the input sequence such that $a'_1 \leq a'_2 \leq \cdots \leq a'_n$.
- **Problem Instance:** An instance consists of the input needed to compute a solution (e.g., the sequence (31, 41, 59, 26, 41, 58)).

Correctness

- **Correct Algorithm:** An algorithm is correct if, for every input instance, it halts in finite time and outputs the correct solution.
- **Incorrect Algorithm:** This may not halt on some inputs or may return an incorrect answer.
- **Note:** Incorrect algorithms can be useful if their error rate can be controlled (e.g., algorithms for finding large prime numbers).

Key Specialized Concepts

- **Data Structures:** A way to store and organize data to facilitate access and modifications. No single structure works for all purposes.
- **NP-Complete Problems:** A subset of problems for which no efficient algorithm has been found, yet it hasn't been proven that one cannot exist. If an efficient algorithm exists for one NP-complete problem, it exists for all of them.
    - *Example:* The **Traveling-Salesperson Problem** (finding the lowest distance for a delivery truck to visit several addresses and return to the depot) is NP-complete.
- **Parallelism:** To handle physical limits on processor clock speeds, multi-core computers use multiple processing cores. Algorithms must be designed with parallelism in mind to be efficient.
- **Online Algorithms:** Algorithms that receive input over time rather than having all data available at the start (e.g.,

job scheduling in a data center or hospital triage).

1.2 Algorithms as a Technology

Even if computers were infinitely fast and memory were free, algorithms would be necessary to ensure solutions are correct and terminate. However, since time and memory are bounded resources, efficiency is vital.

Efficiency Comparison: Insertion Sort vs. Merge Sort

Different algorithms for the same problem can vary greatly in efficiency.

- **Insertion Sort:** Takes time roughly proportional to n^2 ($c_1 n^2$).
- **Merge Sort:** Takes time roughly proportional to $n \lg n$ ($c_2 n \lg n$).
- **The "Crossover Point":** While insertion sort may be faster for small inputs due to smaller constant factors, merge sort's slower growth rate makes it significantly faster as n increases.

Example Case: A fast computer (A) running insertion sort vs. a slower computer (B) running merge sort to sort 10 million numbers.

- **Computer A (10 billion instructions/sec):** Takes ~20,000 seconds (over 5.5 hours).

- **Computer B (10 million instructions/sec):** Takes ~1,163 seconds (under 20 minutes). By choosing a better algorithm, the slower hardware outperforms the faster hardware by over 17 times.

Algorithms and Other Technologies

Algorithms are as much a technology as hardware. They are at the core of:

- **System Infrastructure:** Hardware design, GUIs, and networking.
- **Software Development:** Compilers and interpreters.
- **Machine Learning:** A collection of algorithms that infer patterns from data.
- **Data Science:** Uses algorithmic design and analysis to extract insights from data.

Would you like me to create a set of practice problems or a LaTeX-formatted cheat sheet based on these definitions and the efficiency comparison?