

Basics of Algorithms

Algorithms - Definition

- “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**”
- “A **tool** for solving a well-specified **computational problem**”

Algorithm Example – Sorting Problem

- **Computational Problem:** *Sort a sequence of numbers into non-decreasing order*
- **Input and Output:**

Input: sequence $\langle a_1, a_2, \dots, a_n \rangle$ of numbers.

Output: permutation $\langle a'_1, a'_2, \dots, a'_n \rangle$ such that $a'_1 \leq a'_2 \leq \dots \leq a'_n$.

Example:

Input: 8 2 4 9 3 6

Output: 2 3 4 6 8 9

Algorithms – Real World Scenarios

- Examples of real world scenarios that require the development of algorithms
 - Internet -Finding good routes on which data will travel and using search engines to quickly find pages on which particular information resides
 - DNA Analysis – Human Genome Project
 - Resource Allocation – Optimizing campaign funds
 - GPS Routing
 - Cybersecurity
 - ...

Algorithms – Correctness

- **Definition:** For every input instance, it halts with the correct output. We say that a correct algorithm solves the given computational problem
- An *incorrect algorithm* might not halt at all on some input instances, or it might halt with an incorrect output

Algorithms – Analysis

The theoretical study of computer-program performance and resource usage.

What's more important than performance?

- modularity
- correctness
- maintainability
- functionality
- robustness
- user-friendliness
- programmer time
- simplicity
- extensibility
- reliability

Why study Algorithms and performance?

- Algorithms help us to understand *scalability*.
- Performance often draws the line between what is feasible and what is impossible.
- Algorithmic mathematics provides a *language* for talking about program behavior.
- Performance is the *currency* of computing.
- The lessons of program performance generalize to other computing resources.
- Speed is fun!

Data Structures -Review

- **Definition:** *“A data structure is a way to store and organize data in order to facilitate access and modifications”*
- Elementary Data Structures
 - Arrays and Strings
 - Linked Lists
 - Stacks and Queues
 - Trees and Graphs
 - Heaps
 - Hash Tables