Problem Statement: Implementation of Linear Search Algorithm

Heet Trivedi a.k.a Arjun

August 2025

1. Introduction

The "Abramov's Algorithm" project focuses on computational methods for solving mathematical problems, including algorithms for searching and processing data. As part of extending the project's algorithmic library, this problem statement outlines the implementation of a linear search algorithm across multiple programming languages. Linear search is a fundamental algorithm that sequentially searches for a target value in a list or array, serving as a baseline for understanding search strategies in computational contexts.

2. Problem Description

The objective is to develop a robust and consistent implementation of the linear search algorithm in the following programming languages: C, C++, Java, Python, Go, Rust, Kotlin, MATLAB, Scala, Perl, Ruby, Julia, and Haskell. The algorithm must efficiently locate a target value within an unsorted array or list and return its index, adhering to the project's proprietary restrictions.

2.1. Algorithm Overview

- **Linear Search**: Iterates through each element in an array/list to find a target value, checking for equality at each step.
- **Input**:
  - An array or list of elements (e.g., integers).
  - A target value to search for.
- **Output**:
  - The index of the target value if found (0-based indexing).
  - -1 if the target value is not present in the array/list.
- **Time Complexity**: $O(n)$, where n is the length of the array/list.

- **Example**:
  - Input: Array $[3, 7, 1, 9, 4]$, Target 9
  - Output: 3 (index of 9)

2.2. Requirements

- **Functionality**:
  - Implement a function/method named `linear_search` (or language-appropriate equivalent) that accepts an array/list and a target value.
  - Return the index of the first occurrence of the target or -1 if not found.
- **Input Constraints**:
  - Array/list contains integers (adaptable to other types if needed).
  - Array/list size: $1 \leq n \leq 10^6$.
  - Target value is an integer within the range of array elements.
- **Output Format**:
  - Print a message: "Target `<target>` found at index: `<index>`" (e.g., "Target 9 found at index: 3").
- **Language-Specific Guidelines**:
  - Use idiomatic constructs for each language (e.g., `for` loops in C, `foreach` in Ruby, list comprehensions avoided in Python for clarity).
  - Ensure portability and minimal dependencies (no external libraries required).
- **Test Case**:
  - Use array $[3, 7, 1, 9, 4]$ and target 9 for consistency across implementations.

3. Scope

- **Included Languages**: C, C++, Java, Python, Go, Rust, Kotlin, MATLAB, Scala, Perl, Ruby, Julia, Haskell.
- **Excluded Languages**:
  - **Solidity**: Unsuitable for general-purpose algorithms due to its blockchain focus.
  - **Cirq/Qiskit**: Python-based quantum computing frameworks; their implementation would be redundant with Python's.