

Project Proposal: Personalized Algorithm Practice Platform

Overview

This project proposes developing a personalized algorithm practice platform designed to assist students and learners in improving their proficiency in various algorithmic topics. Utilizing existing question sets (such as NeetCode 250), the platform will dynamically recommend algorithm practice problems based on individual user performance and proficiency, providing tailored learning experiences.

Objectives

- Automatically assess user proficiency in different algorithm categories based on past performance.
- Recommend practice problems intelligently, guiding users toward topics where improvement is most needed.
- Enhance learning through AI-driven explanations, leveraging Large Language Models (LLMs) to analyze submitted solutions and provide insights.

Core Algorithms

1. User Proficiency Scoring Algorithm

- Computes user proficiency scores by analyzing historical data such as correctness rates, completion times, and problem difficulties.

2. Intelligent Recommendation Algorithm

- Recommends algorithm problems dynamically by prioritizing topics with lower user proficiency scores, encouraging balanced skill improvement.

3. Sorting Algorithm

- Used to sort algorithm problems or user statistics by difficulty, performance metrics, or other relevant criteria.

Optional Enhancements:

- Collaborative filtering or content-based recommendation for improved personalization.
- Visualization of proficiency through interactive graphs.

Data Structures

The project will utilize the following core data structures:

- **Hash Tables:** Efficient storage and retrieval of user proficiency data.

- **Priority Queues (Heaps):** Dynamic prioritization in recommending practice problems.
- **Arrays:** Structured storage of the 250 algorithm problems and user performance records.
- **Queues or Stacks:** Supporting algorithm visualization tools or managing sequential tasks (optional).
- **Graphs:** Illustrating relationships among algorithm topics (optional).
- **Linked Lists:** Managing user action histories or recent activities (optional).
- **Binary Search Trees (BST):** Efficiently storing and retrieving sorted data such as user scores or problem difficulties (optional).

This project aligns with the course objectives by demonstrating the practical use of algorithms and data structures to solve a real-world, personalized learning challenge.