Digantara Backend Developer Assignment

Objective:

Design and implement one or more APIs (using language of your choice) that demonstrate your understanding of fundamental data structures and algorithms (DSA). Your solution should include implementations of the following algorithms:

- Binary Search
- Quick Sort
- Breadth First Search (BFS)

In addition, your API(s) must log each call by recording:

- Algorithm Name (which API was called)
- Input Provided to the algorithm
- Output Produced by the algorithm

So that we can revisit the results of previous queries and not have to run the same query multiple times.

Requirements:

API Design:

▼ Single vs. Multiple APIs:

You may choose to implement a single API that handles all three algorithms or separate APIs for each algorithm.

▼ Input/Output:

The method of receiving input (e.g., command line, HTTP request, socket, etc.) is all your choice

The API should provide clear and structured responses for each algorithm call.

Algorithm Implementation:

- ▼ Binary Search:

  Implement the binary search algorithm. Ensure that your implementation works , should have error handling mechanism for incorrect input if binary search cannot be applied.

- ▼ Quick Sort:

  Given an array Implement the quick sort algorithm for sorting arrays or lists.

- ▼ BFS (Breadth First Search):

  Implement BFS for traversing a tree/graph data structures outputting an array or list of the nodes visited given a source or root node.

Logging/Storage:

Logging:

Each time an API is called, log the following:

- Algorithm Name: The name of the algorithm (e.g., "Binary Search", "Quick Sort", "BFS").
- Input Data: The input provided for that call.
- Output Data: The result produced by the algorithm.

Storage Method:

- You can choose any storage mechanism (e.g. file system, database, in-memory structure with persistence, etc.) to maintain these logs.

Implementation Guidelines:

Language Flexibility:

- You can use any single or multiple programming languages to implement your solution.
- The emphasis is on demonstrating your language proficiency, understanding of DSA concepts, and logical thinking while building an application or API.

Modularity & Readability:

- Write clean, modular, and well-documented code. Use comments and README files to explain your design decisions and how to run/test your API(s).

Error Handling:

- Include proper error handling to manage invalid inputs or unexpected behaviors gracefully.

Submission Instructions:

Provide a GitHub link to your solution with clear instructions, with a Read-Me file having:

- How to build/run your solution.
- How to start/initialize your API(s).
- How to send requests (e.g., sample input format or HTTP endpoints).
- How to access the log or storage of the algorithm calls.
- Include examples to demonstrate how to use your API(s).

Evaluation Criteria:

Your submission will be assessed based on the following:

- ▼ Correctness

  Accurate implementation of the algorithms and proper implementation of API end points.

- ▼ API Design

  Clarity and usability of your API(s), including input/output handling.

- ▼ Code Quality

  Cleanliness, readability, modularity, and documentation of your code.

- ▼ Language Proficiency

  Effective use of your chosen programming language(s) to solve the problem.

- ▼ Creativity

  Any additional features or improvements that enhance the functionality or user experience.

Extra points(Optional):

These features are bonus points for exceptional candidates. They are not mandatory and will not negatively impact your review if not implemented

- Deploy the API server
- Dockerize the application

> Good luck, and we look forward to reviewing your implementation!