Principal Software Engineer Technical Assessment

Background

Our company specializes in data exploration, search, and analysis tools that enable users to navigate and utilize a large entity-relationship graph composed of tens of millions of nodes.

Assignment

Your task is to design and implement a simplified version of a graph-based search and recommendation system. This system should demonstrate your ability to work with large-scale data structures, implement efficient algorithms, and create RESTful APIs.

Requirements

1. Design and implement a graph data structure to represent relationships between entities.

2. Create a RESTful API with the following endpoints:
    - Add a new entity to the graph
    - Add a relationship between two entities
    - Search for entities based on specific criteria
    - Generate recommendations for a given entity based on its relationships

3. Implement a basic search algorithm that can efficiently find entities based on given criteria.

4. Design a simple recommendation algorithm that suggests related entities based on graph structure.

5. Include basic error handling and input validation.

6. Provide clear documentation for your API endpoints and any setup instructions.

Constraints

- Use Python for the implementation.
- You may use any standard libraries or popular frameworks (e.g., Flask, FastAPI) for the API implementation.
- For simplicity, you can use in-memory data structures to represent the graph. However, your design should consider future scalability to a database backend.

- Your solution should be able to handle a graph with at least 10,000 nodes efficiently.

Deliverables

1. Source code for your implementation.
2. A README file with:
    - Setup and running instructions
    - Brief explanation of your design choices and algorithms
    - Any assumptions you made
    - Ideas for further improvements or scaling the solution

Evaluation Criteria

Your submission will be evaluated based on:

1. Code quality and organization
2. Algorithmic and resource efficiency
3. API design and documentation
4. Consideration of scalability and future improvements

Please spend no more than 1.5 hours on this assignment. We're more interested in your approach and design decisions rather than a fully polished solution.

Good luck! We look forward to reviewing your submission and discussing it with you in the follow-up session.