Citadel - Algorithm Design Assignment

Background Context

We're building a social networking platform for university students that combines profile discovery (similar to Hinge's approach) with group dining experiences (inspired by Timeleft's concept).

Your Challenge

Design **TWO CORE ALGORITHMS** that will be the backbone of our matching system. You have complete creative freedom in your approach - whether that's mathematical modelling, weighted scoring systems, machine learning approaches, or hybrid solutions.

Algorithm 1: Profile Discovery Engine

The Problem

Users see profiles one by one-by-one and rate them using traditional like/dislike options. After each rating, the system must intelligently decide which profile to show next.

Available User Data Points

- **Academic**: University, degree program, graduation year
- **Geographic**: City, preferred areas within the city
- **Personal**: Age, gender, interests/skills (max 5 selected)
- **Behavioural**: Previous like/dislike patterns, response patterns
- **Profile Content**: Bio text, uploaded media, response to personality prompts

Core Requirements

1. **Prevent Randomness**: The "next profile" decision should be strategic, not random
2. **Progressive Filtering**: Balance between familiar (same university/degree) and diverse (different backgrounds)
3. **Learning Component**: The System should adapt based on users' like/dislike patterns
4. **Engagement Optimisation**: Keep users engaged while maximising meaningful connections

Success Metrics

- Higher mutual positive ratings
- Reduced time to find compatible matches
- Sustained user engagement over time

Algorithm 2: Group Dining Matcher

The Problem

Create balanced 6-person dinner groups where strangers will spend 2+ hours together. The experience should feel natural and engaging for all participants.

Available User Data Points

- **Lifestyle Preferences**:
    - Dietary restrictions (vegetarian, vegan, no restrictions)
    - Budget comfort level (₹500-800, ₹800-1200, ₹1200+)
    - Language preferences
    - Alcohol consumption (yes/no)
    - Relationship status (single, in a relationship, not looking)
- **Interest Indicators**: Taken from the user's profile interests/skills and bio content
- **Social Context**: Social goals and preferences

Core Requirements

1. **Hard Constraints**: Dietary, budget, and location compatibility must be absolute
2. **Interest Balance**: Create groups with complementary interests from user profiles, not identical ones
3. **Conversation Potential**: Ensure sufficient common ground while maintaining diversity
4. **Fairness**: Every user should have equal opportunity for great experiences

Success Metrics

- Post-dinner satisfaction ratings
- Number of lasting connections formed
- Repeat booking rates

Technical Specifications

Data Structures You Can Assume

User Profile:
- demographics: {age, gender, city, university, degree, year}
- preferences: {dietary, budget, languages, alcohol, relationship_status}
- interests: [list of selected interests/skills from profile]
- behavioral_data: {previous_ratings, response_patterns}

Constraints

- **Scale**: 10,000+ active users per city
- **Performance**: Algorithm must execute in <100ms
- **Fairness**: No user should be consistently disadvantaged
- **Privacy**: Minimise data exposure between users

Evaluation Criteria

Algorithm Design (75%)

- **Logic Soundness**: Are your assumptions and reasoning solid?
- **Scalability**: Will this work with thousands of users?
- **Adaptability**: Can the system learn and improve over time?
- **Edge Case Handling**: What happens with limited data or unusual user profiles?
- **Innovation**: Creative approaches that go beyond obvious solutions

Presentation Quality (25%)

- **Presentation:** Code, simulation, wireframe, document, pen and paper, anything works.
- **Clarity**: Can a non-technical stakeholder understand your approach?
- **Completeness**: Have you addressed all requirements?
- **Practical Considerations**: Implementation challenges and solutions
- **Testing Strategy**: How would you validate algorithm performance?

Submission Requirements

1. **Core Algorithms**: Complete logic for both matching systems
2. **Documentation**: Clear explanation of your approach and reasoning
3. **Trade-off Analysis**: What compromises did you make and why?
4. **Implementation Plan**: How would you deploy and monitor these algorithms?

Timeline

Submission Deadline: 30th June
Format: Reply to the email with a drive link, a PDF, or a GitHub link; basically, anything works.

Final Notes

This is your opportunity to demonstrate not just coding ability, but strategic thinking about complex matching problems. We're looking for candidates who can:

- *Think systematically about user behaviour and psychology*
- *Balance multiple competing objectives*
- *Design scalable solutions for real-world constraints*
- *Communicate complex ideas clearly*

Remember: There's no single "correct" answer. We're evaluating your problem-solving approach, creativity, and ability to design systems that create meaningful human connections.

Good luck!