Intro to AI
Tutorial 5

1. What is the worst-case time complexity of Hill Climbing if each state has b neighbors and the search space has N states? How does this change in Steepest-Ascent Hill Climbing?
2. **Problem Scenario**

   You're organizing a 3-day music festival with 5 stages. You need to create the optimal schedule for 40 bands, considering multiple real-world constraints.

   Parameters to Optimize (Genes):

   1. Band Assignments:
      - 40 bands
      - 5 stages
      - 3 days
      - 4 time slots per day (afternoon/evening)

   2. Stage Allocations:
      - Main Stage (capacity: 20,000)
      - Rock Stage (capacity: 10,000)
      - Electronic Stage (capacity: 8,000)
      - Indie Stage (capacity: 5,000)
      - New Artist Stage (capacity: 3,000)

   3. Hard Constraints:
      - No band can play twice in the same day
      - Headliners must play evening slots
      - No two major bands can play simultaneously
      - Each stage must have breaks for setup (30 mins)

   4. Soft Constraints (Preferences):
      - Similar genres shouldn't overlap
      - Popular bands should get larger stages
      - Local bands should play earlier slots
      - Fan-favorite bands should be spread across days

```
band_data = {
    'popularity_score': 1-100,
    'genre': ['rock', 'electronic', 'indie', 'pop'],
    'performance_length': 30-90 mins,
    'expected_audience': number,
    'is_headliner': boolean,
    'is_local': boolean
}
```

Questions:

1. Design a chromosome representation for this schedule. Explain how you would:
    - Encode band assignments
    - Handle time slots
    - Represent stage allocations
    - Ensure feasibility of solutions

2. Your initial population includes this partial schedule:

```
Day 1, Main Stage:
2PM: Band A (Rock, 70k fans)
4PM: Band B (Pop, 50k fans)
6PM: Band C (Rock, 90k fans)
8PM: Band D (Pop, 100k fans)

Day 1, Rock Stage:
2PM: Band E (Rock, 40k fans)
4PM: Band F (Metal, 45k fans)
6PM: Band G (Rock, 55k fans)
8PM: Band H (Metal, 35k fans)
```

Identify all constraint violations and calculate a fitness penalty.

3. Given two parent schedules, demonstrate: a) How would you perform crossover while maintaining schedule validity? b) Design two mutation operators specific to this problem c) How would you handle repair of invalid schedules after genetic operations?
4. For this fitness distribution in a population:

Schedule	Base Fitness	Constraint Violations
1	0.85	2 minor
2	0.78	1 major
3	0.92	3 minor
4	0.70	0
5	0.88	1 minor

Calculate final fitness scores using:

- Minor violation penalty: -0.1

- Major violation penalty: -0.3