You are tasked with building a **flight route planner** for a fictional airline company. This planner will help customers find the **cheapest route** between two cities based on a complex dynamic pricing system.The airline operates between several cities, but flight prices are **dynamic** and depend on various factors, such as the time of booking, distance, availability, and special offers.

**Requirements:**

1. **Graph Representation**:
   * Each city is a **node**.
   * Each flight is an **edge** with a weight representing the current price.
   * The weight (price) between any two cities is **dynamic**, meaning it can change with time.
2. **Dynamic Pricing System**:
   * Prices fluctuate based on:
     + **Time of day** (peak/off-peak hours).
     + **Availability** (more expensive when fewer seats are left).
     + **Distance** between cities.
     + **Special discounts** or promotions applied periodically.
3. **Route Planner**:
   * The user can request the **cheapest route** between two cities. Your algorithm should handle:
     + **Single flight trips**.
     + **Multi-leg trips** (trips with multiple connecting flights).
   * The system should respond quickly, even when prices dynamically change during the day.
4. **Additional Features**:
   * The system should support **round-trip planning** with dynamic pricing on both legs of the trip.
   * Allow **queries with constraints** such as:
     + Max number of connections.
     + Flight duration limits (some flights take longer, e.g., layovers).
5. **Efficiency**:
   * The system should handle **large datasets** with up to **1,000 cities** and **10,000 possible flight routes**.
   * Consider using advanced data structures for faster lookups and updates.

**Example list of cities and possible flights:**

City\_A, City\_B, Distance = 500 km

City\_B, City\_C, Distance = 300 km

City\_A, City\_C, Distance = 800 km

**Pricing function logic:**

price = base\_price + dynamic\_factor

base\_price = Distance \* 0.1 (e.g., $0.10 per km)

dynamic\_factor = f(time, availability, promotions)

**User queries:**

Find the cheapest route from City\_A to City\_C

Find a round-trip from City\_A to City\_B with at most one connection.