

Intuition:

The Intuition is to count all possible routes from a start city to a finish city, given a certain amount of fuel. It uses a recursive approach with memoization to optimize the computation.

The helper function performs the main computation. It checks if the remaining fuel is sufficient for each possible move and recursively explores all valid paths. The results of subproblems are stored in a memoization table to avoid redundant computations.

By using memoization, the code avoids recomputing the same routes multiple times, significantly improving efficiency. The approach enables the algorithm to handle larger inputs by reusing previously computed results, reducing the overall time complexity.

Explanation:

1. The helper function is a recursive function that performs the main computation for counting all possible routes. It takes the following parameters: locations (vector of city locations), city (current city index), finish (index of the finish city), remainFuel (remaining fuel), and memo (memoization table).
2. The function begins by checking if the remaining fuel (remainFuel) is negative. If so, it means that the current path is not feasible due to insufficient fuel, and the function returns 0.
3. Next, the function checks if the result for the current city and remainFuel combination is already present in the memo table. If so, it retrieves and returns the stored result, avoiding redundant computations.
4. If the result is not available in the memo table, the function proceeds to compute it.
5. The function initializes the total variable as 1 if the current city is the same as the finish city, indicating that a valid route has been found. Otherwise, total is initialized as 0.
6. The function then iterates over all possible nextCity indices, representing the cities that can be visited from the current city. It skips the current city itself.
7. For each nextCity, the function checks if there is enough fuel (remainFuel) to travel from the current city to the nextCity. If so, it recursively calls the helper function for the nextCity with the updated remainFuel (subtracting the fuel consumed for the current move) and adds the returned result to the total value.
8. Before adding the result to total, the function takes modulo MOD to keep the result within the specified range (1000000007).
9. Once all possible nextCity options have been explored, the function stores the computed total result in the memo table for the current city and remainFuel combination. This is done to avoid redundant computations in future recursive calls.
10. Finally, the function returns the computed total result.