

**CSC 114 Algorithms Analysis and Design**

**to:**

**Dr. Ahmed Bayoumi**

**Travel Recommendation System Documentation**

|  |  |
| --- | --- |
| **Name** | **Mazen Mahmoud Abdelhamid** |
| **ID** | **320210301** |
| **Group** | **5** |
| **Section** | **14** |

**Overview:**

The Travel Recommendation System is a Python-based application that recommends travel destinations and hotels based on budget constraints. It utilizes concepts from the knapsack problem, dynamic programming, and backtracking to provide optimized recommendations.

**Features:**

- Recommends a specified number of cities to visit based on the user's current location.

- Considers average flight prices to each destination, taking into account both outgoing and return trips.

- Filters hotels based on the selected cities and calculates the average hotel prices.

- Optimizes the selection of cities and hotels within the given budget constraint.

- Provides the recommended cities along with the total budget required for flights and hotels.

**Dataset:**

The system uses three datasets:

1. `users.csv`: Contains user preferences including user codes and current location.

2. `flights.csv`: Includes flight information such as travel codes, destinations, prices, and dates.

3. `hotels.csv`: Consists of hotel data including place, days, prices, and total.

**Algorithm:**

The algorithm used in this code involves concepts from the knapsack problem, dynamic programming, and backtracking.

1. Knapsack Problem:

The knapsack problem is a classic optimization problem that involves selecting items with maximum value while staying within a given weight or capacity constraint. In this code, we adapt the knapsack problem to select cities and hotels within a given budget constraint.

2. Dynamic Programming:

Dynamic programming is a technique for solving complex problems by breaking them down into overlapping subproblems and solving each subproblem only once, storing the results for future reference. In this code, dynamic programming is used to solve the knapsack problem by calculating the maximum number of cities that can be visited within the budget for each city and budget combination.

3. Backtracking:

Backtracking is a technique used to find solutions by exploring all possible paths and backtracking when a solution is not feasible or optimal. In this code, backtracking is used to find the selected cities and total budget by tracing back from the last city and budget combination to the first, based on the calculated dynamic programming table.

The algorithm works as follows:

1. Load the travel dataset and user preferences.

2. Filter flights based on the user's current location and calculate the average flight prices to each destination.

3. Sort the destinations based on the average flight prices and select the unique cities with the lowest prices.

4. Create a dynamic programming table and a budget table to store the maximum number of cities and total budget for each city and budget combination.

5. Iterate over each city and budget combination, considering the flights and hotels. Calculate the maximum number of cities that can be visited within the budget using dynamic programming, updating both tables.

6. Backtrack from the last city and budget combination to the first, finding the selected cities and total budget based on the dynamic programming table.

7. Return the recommended cities along with the remaining budget.

**Test Cases:**

1. Enter The User Name : Scott Evans

Enter Your Budget : 10000

Enter Number of Cities You Want To Visit : 5

Enter Your Location : Florianopolis (SC)

Output:

Recommended cities and hotel prices: [('Rio de Janeiro (RJ)', 165), ('Sao Paulo (SP)', 139), ('Campo Grande (MS)', 60), ('Brasilia (DF)', 247), ('Recife (PE)', 312)]

Total budget: 5024

**2-** Enter The User Name : Kathleen Heller

Enter Your Budget : 3000

Enter Number of Cities You Want To Visit : 2

Enter Your Location : Campo Grande (MS)

Output:

Recommended cities and hotel prices: [('Sao Paulo (SP)', 139), ('Natal (RN)', 242)]

Total budget: 1849

**3-** Enter The User Name : Carl Branch

Enter Your Budget : 2000

Enter Number of Cities You Want To Visit : 2

Enter Your Location : Brasilia (DF)

Output:

Recommended cities and hotel prices: [('Rio de Janeiro (RJ)', 165)]

Total budget: 1042

**Conclusion:**

The Travel Recommendation System provides an efficient and optimized solution for suggesting travel destinations and hotels based on user preferences and budget constraints. By leveraging concepts from the knapsack problem, dynamic programming, and backtracking, it provides valuable recommendations to enhance the travel planning experience.