CPSC 335 - Algorithm Engineering
Project 1 Report
Spring 2025
Due: 02/28

Wayne Muse
88601211
CSUF Email**:** waynemuse@csu.fullerton.edu

Marck Villatoro
CSUF Email: marckvillatoro@csu.fullerton.edu

Submission for Project 1

Introduction

This project involves the implementation and analysis of two algorithms:

1. **Greedy Approach to the Hamiltonian Problem** – Determines the best starting city for a circular route based on available fuel and distances.
2. **Connecting Pairs of Persons** – Computes the minimum number of swaps needed to seat couples together.

Both algorithms have been implemented in Python and analyzed for their time complexity of O(n).

Algorithm 1: Greedy Approach to Hamiltonian Problem

Problem Description

Given a circular route of cities where each city provides a certain amount of fuel, the goal is to determine the starting city such that a vehicle can complete the full circular journey without running out of fuel. The vehicle consumes fuel at a given miles-per-gallon (MPG) rate.

Implementation Details

- The algorithm iterates through the list of cities, tracking total fuel and total distance traveled.
- If the current fuel tank balance becomes negative at any city, the next city is set as the new starting point.
- After one full traversal, the algorithm checks whether the total fuel is sufficient to complete the journey. If so, it returns the starting city; otherwise, it returns -1.

Time Complexity Analysis

- The algorithm iterates through the list of cities once (O(n)).
- Each lookup and arithmetic operation is performed in constant time (O(1)).
- Overall time complexity: O(n).

Example Execution

```
Choose which algorithm question
Type 1 for Algorithm 1: Greedy Approach to Hamiltonian Problem
Type 2 for Algorithm 2: Connecting Pairs of Persons
Type 3 to Exit
1

=====================================
Greedy Approach to Hamiltonian Problem

City Distances: 5 25 15 10 15
Fuel: 1 2 1 0 3
Mpg: 10
Starting City is 4

Choose which algorithm question
Type 1 for Algorithm 1: Greedy Approach to Hamiltonian Problem
Type 2 for Algorithm 2: Connecting Pairs of Persons
Type 3 to Exit
3

=====================================
               Exiting

PS C:\Users\wmuse\OneDrive\Documents\GitHub\CPSC-335-Project1>
```

Algorithm 2: Connecting Pairs of Persons

Problem Description

Given a row of individuals represented by integers, where couples are identified by consecutive numbers (e.g., (0,1), (2,3)), the goal is to compute the minimum number of swaps required to ensure every couple is seated together.

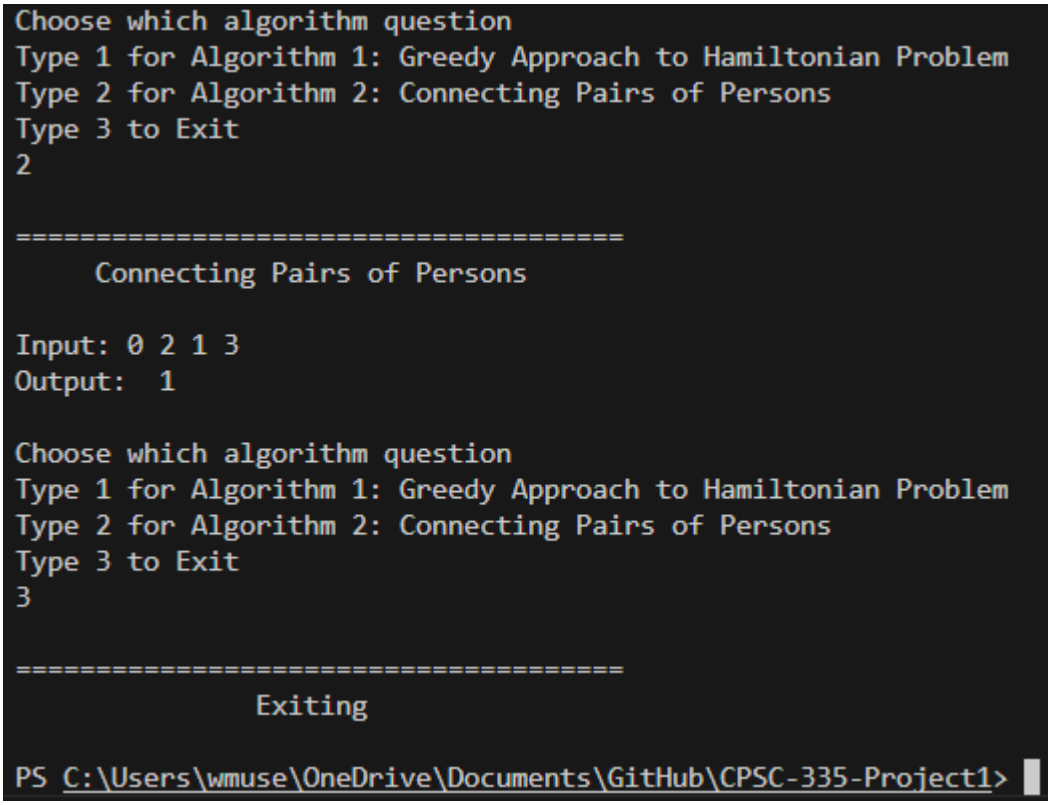
Implementation Details

- A dictionary is used to map each person's position.
- The algorithm iterates through the row in pairs, checking if each pair is a valid couple.
- If not, a swap is performed to place the correct partner together, and the mapping is updated accordingly.

Time Complexity Analysis

- Constructing the position dictionary takes O(n).
- Iterating through the row takes O(n), and each swap operation is O(1).
- Overall time complexity: O(n).

Example Execution



Project Components

The project submission includes the following files:

- **README.md** – Contains instructions for running the project.
- **main.py** – The main driver program to execute the algorithms.
- **alg1.py** – Implementation of the Hamiltonian Problem solution.
- **alg2.py** – Implementation of the Connecting Pairs of Persons solution.

Conclusion

This project successfully implemented and analyzed two algorithms with O(n) time complexity. The solutions were tested with sample inputs and provided correct results. The implementation follows best coding practices with appropriate documentation and structured logic.