Assignment 4: The Harry Potter Reunion (10%)

Start Date: 21st March 2025
Submission Deadline: 5th April 2025 (11.59 PM)
Submission Medium: D2L

Background and Storyline:

The magical world is buzzing with excitement as Harry Potter, Hermione Granger, Ron Weasley, and other Hogwarts alumni plan a grand reunion in Canada. Each alumnus is currently in a different province across the country, and magical paths connect these locations. The reunion is set to take place in Ottawa, the capital of Canada, which serves as the perfect central meeting point for everyone. However, the journey is fraught with challenges such as long distances, travel time, and even dangerous Dementor infestations! As the alumni are scattered across various provinces, it's up to you to help them find the best magical paths to gather in Ottawa.

Your task is to write a Python program that identifies the optimal routes based on four key factors: the shortest hop path (fewest connections), the shortest distance path, the shortest time path, and the path with the least exposure to Dementors. Your program must calculate the most efficient routes, ensuring that each alumnus arrives safely and on time, navigating around the Dementors and other obstacles along the way.

Here are the alumni and their current locations:

- Harry Potter: British Columbia
- Hermione Granger: Ontario
- Ron Weasley: Quebec
- Luna Lovegood: Newfoundland and Labrador
- Neville Longbottom: Saskatchewan
- Ginny Weasley: Nova Scotia

Objectives:

You will:

1. Represent the provinces and magical paths as a graph:
   - Model the provinces as nodes and the magical paths between them as edges in a graph.
2. Calculate four types of optimal paths:
   - Shortest Hop Path (SHP): Find the path with the fewest connections (edges) between provinces.
   - Shortest Distance Path (SDP): Identify the path that minimizes the total distance between the starting province and Ottawa.

- Shortest Time Path (STP): Determine the path that minimizes the total travel time.
- Fewest Dementors Path (FDP): Find the path that exposes the traveler to the least number of Dementors.

3. Implement at least 2 appropriate pathfinding algorithms:
   - You can use Depth First Search (DFS), Breadth First Search (BFS), Dijkstra's Algorithm, or any other suitable algorithms to calculate the optimal paths for each criterion.
4. Provide clear output for the optimal paths for each criterion:
   - Ensure that your program outputs the best path according to each metric (SHP, SDP, STP, FDP) in a clear and understandable format.
5. Visualize the magical transportation network:
   - Use Python libraries like matplotlib or networkx to create a graphical representation of the magical paths between provinces.

Table of Magical Paths (hypothetical)

Starting Province	Destination Province	No. of Hops	Distance (km)	Time (hrs)	Dementors
British Columbia	Saskatchewan	13	1800	19	6
Alberta	Quebec	3	2000	21	7
Ontario	Nova Scotia	2	1300	13	4
Quebec	Newfoundland and Labrador	13	1900	20	26
Nova Scotia	Saskatchewan	2	1800	18	5
Alberta	Saskatchewan	6	1600	8	3
Newfoundland and Labrador	Alberta	4	2400	24	9
Ontario	Quebec	10	500	5	1
Nova Scotia	Ontario	3	2000	21	7
Saskatchewan	Nova Scotia	3	2000	20	37
Quebec	Saskatchewan	4	200	2	0
Alberta	Ottawa	3	2400	24	9
Saskatchewan	Quebec	2	2000	20	6
Ontario	Alberta	2	1500	16	4
British Columbia	Saskatchewan	2	1200	14	3
Newfoundland and Labrador	Quebec	3	2200	22	7
Nova Scotia	Newfoundland and Labrador	10	1200	12	6
Quebec	Ottawa	29	1800	19	17
Alberta	British Columbia	2	1800	18	27

British Columbia	Quebec	2	1900	19	7
Ontario	Newfoundland and Labrador	3	2300	23	8
Nova Scotia	Alberta	3	2200	22	8
Newfoundland and Labrador	Alberta	3	2300	23	8
Alberta	Newfoundland and Labrador	3	2400	24	9
Saskatchewan	British Columbia	3	2000	21	8
Ontario	Saskatchewan	2	1600	16	5
Quebec	Nova Scotia	2	1000	10	2
Newfoundland and Labrador	Saskatchewan	4	2200	23	19
Nova Scotia	Quebec	2	1100	11	2
British Columbia	Newfoundland and Labrador	4	2500	26	10
Ontario	Ottawa	7	1450	4	12
Alberta	Saskatchewan	5	600	8	3
Quebec	Alberta	2	1700	17	6
Saskatchewan	Nova Scotia	9	1800	18	5
Alberta	Quebec	6	2000	21	6
Nova Scotia	British Columbia	4	2500	26	10
Ontario	Nova Scotia	12	1300	13	4
British Columbia	Saskatchewan	13	1800	19	6

Evaluation Criteria (Grading Rubric)

The assignment will be graded out of 30 points with an additional 5 bonus points available for extra credit.

1. Core Functionality and Demo (16 points):
- Calculates four types of optimal paths: 4*4= 16 points

2. Two types of appropriate pathfinding algorithms (4*2= 8 points)

3. Documentation (6 points):
- Graphs: 2 points
- Documentation: 4 points

4. Extra Credit (Optional, 5 bonus points):
- Determine the optimal path for each alumnus that minimizes all four criteria. (Hint: Normalize the values and formulate an optimization problem, then solve it to achieve this) (excluded from the total of 30 points).

Submission Guidelines

1. Code Submission: Submit the complete code with appropriate comments in D2L.
2. Demo Submission: Submit a screen recording in D2L.
3. Documentation: Submit a PDF report and the graphs in D2L explaining your resultant paths in each criteria and what methods did you use to compute them.

Deliverables

1. Fully functional code (.py).
2. The generated graphs
3. A report detailing your method implementation, challenges faced, and result explanations
4. A video demo of the running code in D2L **(Your code should be executed in the command prompt, ensuring that your computer name is displayed as your identifier to verify that the work is genuinely yours:**

).