Assignment #2: Beam Search

Course Instructor: Dr. Shahnawaz Qureshi
Course: Artificial Intelligence

Due: 9th October 2023, Monday 11:59 p.m.

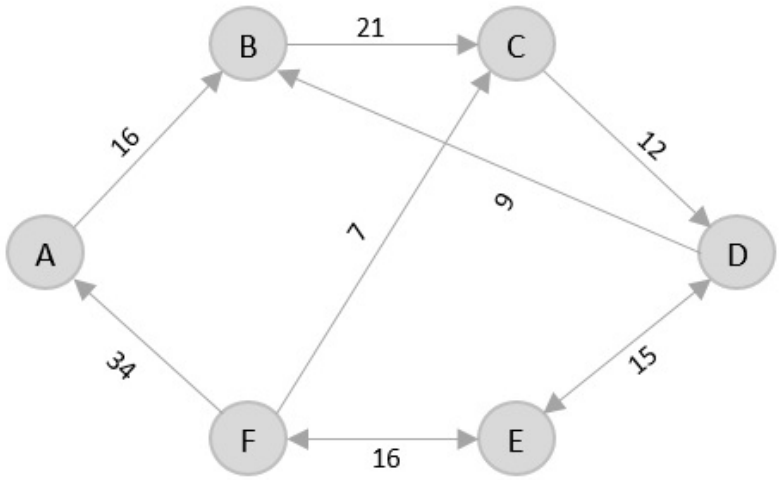
Instructions:

- You are allowed to discuss this assignment verbally but a plagiarism check will be maintained.
- If there are any assignment related queries, please first watch the video posted along with this document. If your query is still unresolved, you can e-mail one of the TA's.
- Submit your file under the naming **convention iXXXXXX_Section.ipynb (E.g. i210328_A.ipynb).**
- Do not zip your files. A 5% penalty will be applied :(.
- Submit the assignment on time.
- Late submissions will receive a 10% penalty.
- Assignment will be due by **9th October 2023, Monday 11:59 p.m.**
- Deadline will not be extended this time. Trust me, you will need the breathing room for the next assignment. This one is easy anyways.
- Lastly, have fun :). The goal with this assignment to create a searching algorithm on an actual real world application. Q2 will stress test your code so I hope it will encourage optimized coding habits.

Assignment #2: Traveling Salesman Problem (TSP)

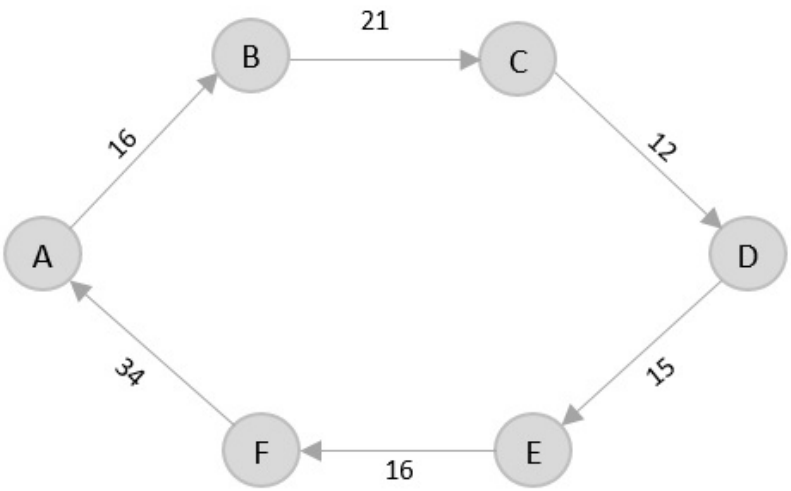
Introduction: The Traveling Salesman Problem is usually defined by the question: "If there is a list of cities and distances between each pair of cities, what is the shortest route that visits each city and returns to the original city?"

Example: In the graph below, if the Traveling salesman starts at 'A', he needs to visit 'B', 'C', 'D', 'E', 'F' and comes back to 'A' while making sure the total distance he has travelled is at a minimum.



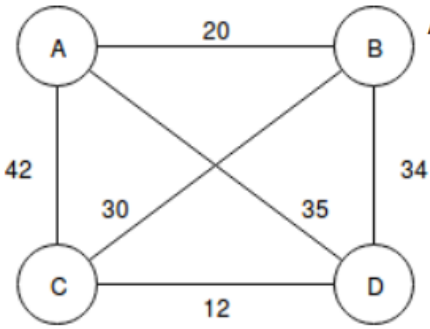
The shortest path that originates and ends at A is A → B → C → D → E → F → A

The cost of the path is: 16 + 21 + 12 + 15 + 16 + 34 = 114.



General layout of TSP

Representation of the Problem



A TSP can be modelled as an undirected weighted graph:

- cities = vertices
- paths between cities = edges
- distance of a path = weight of an edge

This graph can be represented as an **Adjacency matrix**:

\	A	B	C	D
A	0	20	42	35
B	20	0	30	34
C	42	30	0	12
D	35	34	12	0

There are several ways to solve this problem. Namely: Greedy Approach, Naive Approach, Dynamic Programming Approach.

In this assignment, you will be tasked to solve the TSP using **Greedy Approach (Beam Search Variation).**

Assignment Scenario:

A person in the USA has decided to travel from New York to various cities across the USA and come back to his home in New York in 1 trip. He wants to avoid visiting the same city twice (except New York which is his starting and ending city).

With rising fuel prices every day, he wants to cover the least distance possible.

Using the Beam search algorithm, your goal will be to find the shortest possible path for the person starting from New York City, visiting every city afterward only once and comes back home to New York City. A fixed beam value will be provided for Q1 below.

An estimated distance matrix alongside a list of city names w.r.t to their indices is provided **in the Google Colab notebook along with the list of cities.** The indices of the distance matrix correspond to the city names (Refer to the notebook for the code and a more detailed description).

Q1: Using the Beam search algorithm, find the shortest possible path for the person starting from New York City, visiting every city afterward only once and comes back home to New York City. Assume a beam value of 2. For your heuristics, use the total estimated distance which has been travelled thus far by the node. (70 marks)

I.e. if a node has travelled from New York -> Boston -> Chicago so far. It's heuristic value would be 213 + 851 = 1064 Heuristic Value. Your output should show the result in the following format (Where City1 would be replaced by New York, City2 would be replaced by Los Angeles etc. Refer to the notebook for more details)

Path: City1 -> City2 -> City3 -> .... -> City1.

Total Estimated Distance: 2098 miles

Q2: For every Beam value between 2 and 1000 inclusive. Output the beam value, along with path and total distance in the following format. Your starting and ending city should be New York (10 marks):

--- Beam Value: 2 ---

Path: City1 -> City2 -> City3 -> .... -> City1.

Total Estimated Distance: 2098 miles

--- Beam Value: 3 ---

Path: City1 -> City5 -> City2 -> .... -> City1.

Total Estimated Distance: 1578 miles

...

--- Beam Value: 1000 ---

Path: City1 -> City4 -> City3 -> .... -> City1.

Total Estimated Distance: 1448 miles

Q3: Using Matplotlib, draw a simple line plot with Beam Values (2-1000 inclusive) on the X axis and Total Distances w.r.t each Beam Value on the Y axis. Write down your observations below the graph on the best Beam value in this range and its justification (10 Marks).

Q4 In beam search algorithms, how does stochastic beam search help in avoiding local minima solutions? Explain with an example. (10 Marks)