Optimized Flight Trip

Introduction:
This Project explores the application of AI searching algorithms e.g. BFS-DFS-A*-Greedy-UCS-IDS-DLS in optimizing flight trip planning. By harnessing the power of AI, we aim to develop a system capable of efficiently navigating complex flight networks to identify the most cost-effective and time-efficient routes. This project not only delves into the technical aspects of AI algorithms but also examines their real-world implications for travellers, airlines, and the broader travel industry.

CCE414- Artificial Intelligence

Supervised By:

Prof.Dr. Lamiaa Elrefaei.

01

Problem Formulation

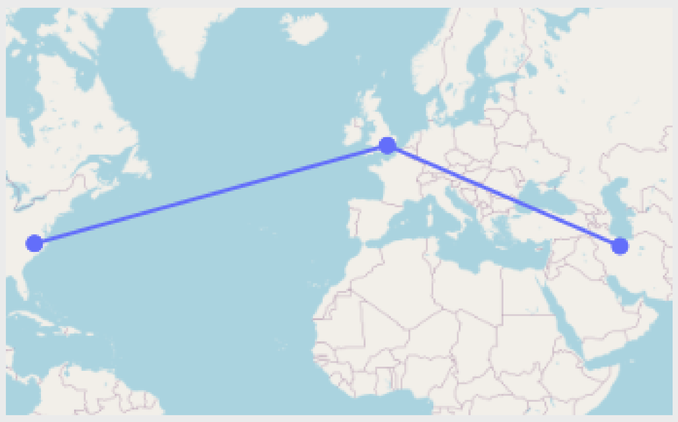
1- Initial State:
'Imam Khomeini International Airport'.
2- Actions:
IN(SOURCE_AIR_PORT):
GO{DESTINATION_AIR_PORT}.
3- Transition Model:
RESULT (IN: SOURCE_AIR_PORT,
GO: DESTINATION_AIR_PORT) =
IN {CURRENT_AIRPORT}
4- Goal Test:
Is CURRENT_AIRPORT = 'Raleigh Durham International Airport'.
5- Path Cost

02

Search Algorithms

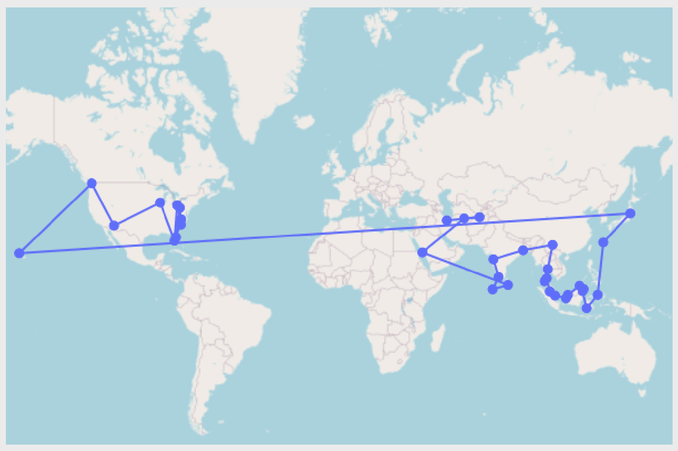
1- Breadth First Search: 0.08027s

BFS is an algorithm that explores a graph level by level. Starting at the root (or an initial node), it systematically explores all neighbors at the present depth before moving on to nodes at the next depth level.



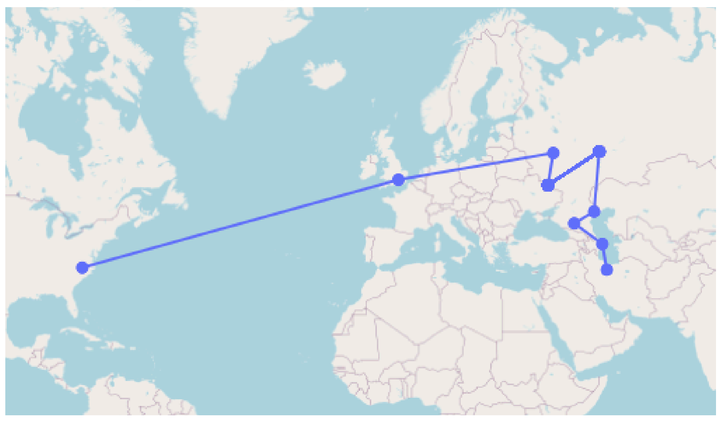
2- Depth First Search: 0.62034s

DFS is an algorithm that explores a graph by going as deeply as possible down one path before backing up and exploring another path.



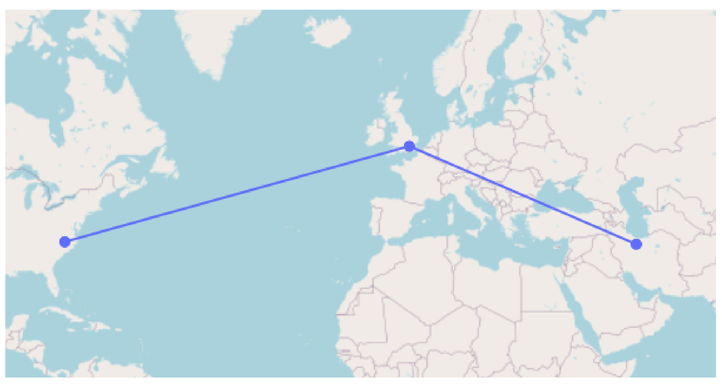
3- Depth Limited Search: 0.01248s

DLS is a variant of DFS where the search is limited to a specified depth. It avoids infinite loops by not expanding nodes beyond a certain depth limit.



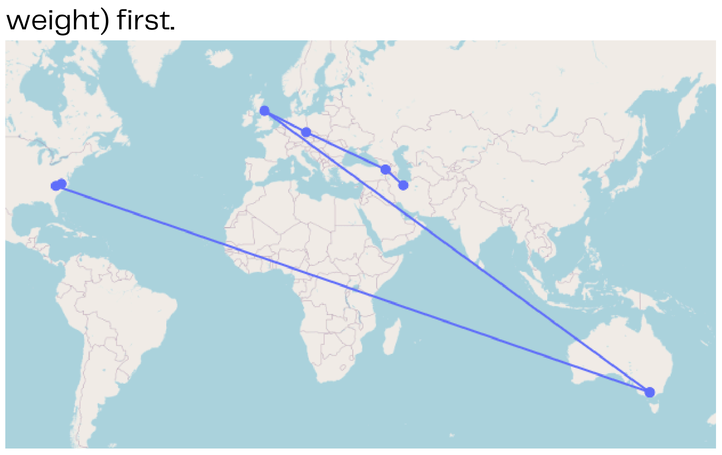
4- Iterative Deepening Search: 0.00519s

IDS is an algorithm that combines the benefits of BFS and DFS. It performs a series of DFS with increasing depth limits until the goal is found.



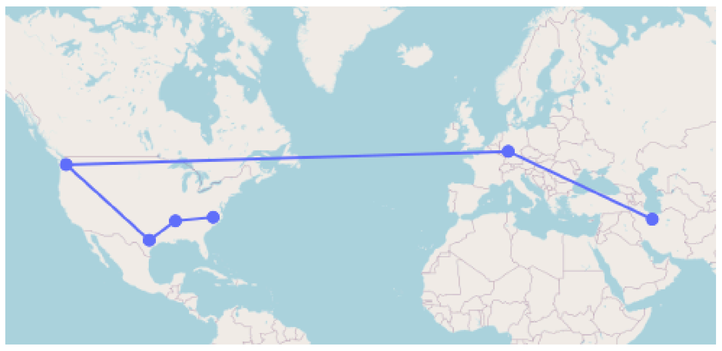
5- Uniform Cost Search: 4.09291s

UCS is an algorithm used for traversing a weighted graph. It expands the node with the lowest cost (path weight) first.



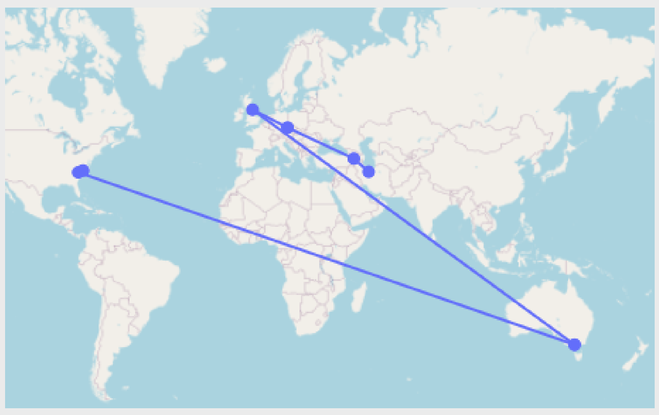
6- Greedy Search Algorithm: 0.20588s

Greedy search is an algorithm that chooses the best immediate option at each step, based on a given heuristic, aiming for a global optimum. It's quick and simple but may not find the best solution, as it doesn't consider future outcomes.



7- A* Search Algorithm: 3.76404s

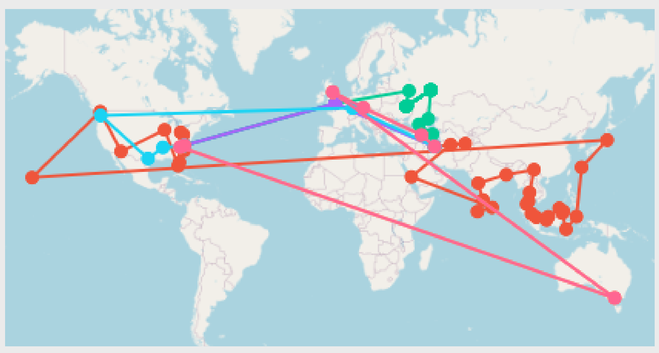
A* search is a pathfinding algorithm that evaluates paths using a combination of the cost to reach a point ('g') and an estimated cost to the goal ('h'). It prioritizes paths with the lowest 'f' score (where f = g + h) to efficiently find optimal solutions, given an admissible and consistent heuristic.



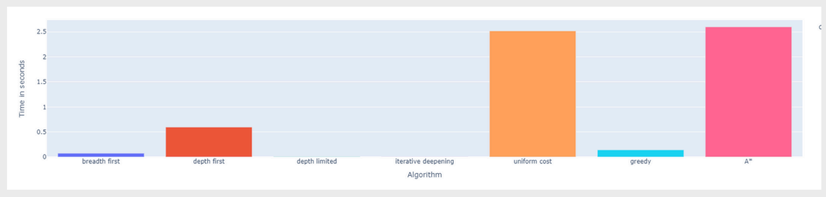
03

Comparison

1- Different Paths:



2- Execution Time:



04

Web Page