

JULIUS NIYONZIMA

461

Professor Lowe

Project overview:

My project is about understanding the pathfinding algorithm, which analyzes five search algorithms to estimate their efficiency in finding the city. I utilized Matplotlib for visualization. The program includes coordinates csv which stores cities and their coordinates, adjacencies.txt which stores cities and their neighbors

The Cities and Coordinates Graph representation:

Cities are labeled as nodes, and roads are labeled as edges, providing connections between the two cities.

I calculated the distance between two cities using the haversine formula for geographical accuracy

Data structures used:

Adjacency list was used to store connected cities

Priority Queues were used in informed search algorithm to prioritize promising routes to find the fastest route to city

Stack and Queues were used by an uninformed search algorithms

The following are the names of **Pathing Finding Algorithm**

Breadth-First-Search:

It works by using queues to exploring the nodes of the graph in order of their distance.

It first begins by exploring the neighbors of the starting city before search further.

Depth -First-Search:

It used stack data structure to explore the cities as far as possible before backtracking until it finds the destination

Iterative-Deepening-Depth_First_Search:

It uses both Breadth-First Search and Depth-First Search. It performs DFS with a depth limit until the goal is reached

Best_First_Search(Informed search).

The best first search use an evaluation function to find the promising path before exploring further.

It uses a priority queue or heap to store the promising paths

A_STAR Algorithm

It combines the breadth_first_search thoroughness and the intelligence of best_first_search to find the fastest and best route.

The combines the cost covered and the estimated cost to the destination to find the optimal path.

The optimal path is guaranteed if the heuristic is admissible.

Requirements for this Program to run

You need to have matplotlib install in python

You need to use python version 3.6 and above for better visualization

You need to import time, pandas, math, and collections

How TO RUN:

The data should be in the same script

After Running the program, it will give you details on how cities are loaded and any city missing in the adjacency list.

It will prompt the user to enter the starting city and destination cities. The cities should be the ones stores in the files listed

Then you have option to view the traversal path of all algorithms or select the the one for best visualization

It provides execution time of each algorithm in microseconds

My interaction with chatGPT

These are the question for my starting point to being writing the program

Calculate the distance between two cities given the latitude and longitude

How can I iterate through a CSV file with cities and coordinates, storing cities, longitudes, and latitudes separately

Create the breadth first to analyze cities with their neighbors

Create the A star to search through cities and their neighbors

Create A depth first search to search from cities

How do I use the coordinates while plotting the graph

How to track memory usage for each algorithm

Create visualization with matplotlib analyzing plotting the cities being traversed by a-star

How do I create a function to analyze the time in microseconds for a-star, best first search, depth first, breadth first, and id-dfs algorithm while the user is selecting the algorithm he wants to analyze

Video link for demonstration

https://drive.google.com/file/d/1pGs2YB2hywQfUnIHirjWFOgoz5TOI-BO/view?usp=drive_link

Work cities

[Haversine formula to find distance between two points on a sphere - GeeksforGeeks](#)

[Best First Search \(Informed Search\) - GeeksforGeeks](#)

[Iterative Deepening Search\(IDS\) or Iterative Deepening Depth First Search\(IDDFS\) - GeeksforGeeks](#)

[A* Search Algorithm - GeeksforGeeks](#)