Yonatan Golan, ID: 206387383

** NOTES:

1. For short paths (~5 steps) the program will run something like 3 minutes until finding a route.
2. The paths will be printed only after all of them (or none) will be found, so if a path was found at first it will be printed only when the last path was found (or will be "No path found")

<u>Problem 1:</u> Chosen Heuristic

My heuristic uses an API I found online, in which I connect to a website named "nominatim.openstreetmap.org".

To this website I send each time I visit a node I consider visiting in my A* implementation the county_id (<county name>, <state code>) and I get back the geographic coordinates (latitude and longitude) of the value I sent.

Afterwards I calculate, using "geopy" library, the geodesic distance in KM between two given coordinates that represent the counties I am considering exploring.

Obviously to perform A* optimally I must have a heuristic that is admissible and consistent.

My heuristic is admissible because geodesic distance is the shortest path between two points on Earth's surface, taking the Earth's curvature into account. So, if the shortest way between A and B is the straight line between them, it will be equal to the heuristic value.

My heuristic is consistent because it satisfies the triangle inequality.

<u>Problem 2:</u> Search

If I have linked list where all node has only one child, where the goal state is the last node (the leaf)

DFS: will iterate n times until finding the last node, which is the goal, so the complexity will be O(n)

IDS: will iterate n times, in which in each iteration it will expand only the last visited node. So it will be like this:

First iteration: Root -> 1st child == 1 visits

Second iteration: Root -> 1st child -> 2nd child == 2 visits

Third iteration: Root -> 1st child -> 2nd child -> 3rd child == 3 visits

$$\sum_{i=1}^{n} i = \frac{1}{2}n(n+1) \rightarrow O(n^2)$$

And that goes on.

The meaning is that every iteration the IDS check (n-1) nodes, as for the n'th iteration it will check n-1 nodes. That's leading to a complexity of O(n^2)