



PROJECT

Implement Route Planner

A part of the Intro to Self-Driving Cars Program

PROJECT REVIEW

CODE REVIEW

NOTES

SHARE YOUR ACCOMPLISHMENT!  

Meets Specifications

Greetings Student,

It was a pleasure to review a good job. The submission is great. Many tips are left to guide you to improve your work.
Thank you, for this good job.

Correctness

Running test.py shows "all tests pass".

All the tests are passed. Nice work!

All tests **pass**! Congratulations!Student's `shortest_path` method implements A* search.The `shortest_path` method perfectly implements A * search in the `student_code` file. Excellent work!

Tips

Some additional materials that talk about the implementation of A* search.

- [A* search algorithm](#);
- [Implementation of A*](#);
- [GeeksforGeeks: A* search algorithm](#).

The heuristic function used to estimate the distance between two intersections is guaranteed to return a distance which is less than or equal to the true path length between the intersections.

The `heuristic` function used to estimate the distance between two intersections perfectly return a distance which is less than or equal to the true path length between the intersections.

Tips

Here are some topics that talk about the distance between two nodes:

- [Optimal way to calculate all nodes at distance less than k from m given nodes](#);
- [Find distance between two nodes of a Binary Tree](#).

Choice and Usage of Data Structures

Code avoids obvious inappropriate use of lists and takes advantage of the performance improvement afforded by sets / dictionaries where appropriate. For example, a data structure like the "open_set" on which membership checks are frequently performed (e.g. `if node in open_set`) should not be a list.

Great! The code uses appropriate structures such that sets/dictionaries. Keep it up!

Comments

A set in Python is a collection of objects without repetition and without order (thus without numbering). This is NOT a sequence!

This item is a judgement call. Student code doesn't need to be perfect but it should avoid big performance degrading issues like...

...unnecessary duplication of lists

...looping through a large set or dictionary when a single constant-time lookup is possible

Great! There is a good code formatting in this submission.

- The code avoids unnecessary duplications of lists;
- And, loops do searches with optimal time on sets.

 [DOWNLOAD PROJECT](#)

[RETURN TO PATH](#)

[Student FAQ](#)