The underlined word with the number must be erased to complete

Else need to check or modify carefully

**Removing the use of complete search (nhớ ghi)**

**1. Presentation of the subject**

If you are a business or organization that has delivery drivers or delivers  
goods or services to many clients, then this applies to you. Couriers, food delivery services, field service businesses, florists, and many more have the same **route planning** problem: how to most efficiently plan, create a map and optimize routes to save time and also resources.

Our aims of this project are…..

**2. Description of the problem**

* Similar the route planning, the intelligent vehicle can only travel  
  from the current city to the cities near it within a certain time, and also additional time if it goes at peak hour. The objective is to try to minimize the time cost to travel between the two cities.
* Formulation:

+ Initial state: In starting city

+ Action/Transition model: Action (In: starting city) = {Go: adjacent cities}

+ Goal test: {In: Goal city}

+ Path cost: Sum of distance

(May add a graph containing 63 nodes instead of 63 cities in Vietnamese)

**3. Algorithms for the problem (some comments)**

Explaining a bit about the UCS function

The A\* star function is similar to the UCS but the heuristic, which is a matrix having the air distance between each of two cities. Unlike the UCS which randomly picks the adjacent city, A\* will take the nearest air distance city from the current city as a priority choice, and the air distance will be added to the heuristic function. The rest cities are like the UCS.

**4. Performance comparison between algorithms (graph, analysis, comment)**

**5. Problems occurring during working on the assignment and how we handled them**

* Input data takes too much time to make. Because of the approach in which we attempt to solve the problem, we need to motivate the team and try hard to input data by hand including creating a distance matrix between cities, and an additional time cost matrix with google support so it can be as accurate as possible.
* Additional time is hard to implement. Our based knowledge is not too much, so we struggle to learn or find a function that can predict traffic time. At last, we came to a simple code, trying to add time costs when traveling at a certain time. Not the best but the only solution we find.
* Call API of google is not available in Vietnam. At first, we don’t know what to do because we only know and are familiar with google maps. However, then we know and learn to use the apple map API but it is a bit annoying because you need the city coordinate to call it precisely.

**6. Conclusion and possible extensions**

**Conclusions:**

* Although the time complexity of the function is O(n2), the n is constant so can use matrix input here
* The heuristic is not as far as good. It can give the optimal time and use less memory if the path connecting cities is straight but the more complex it is, the less effective the A\* function is, and the better UCS performance is in time cost.

**Maybe two more**

**Possible extension:**

* Try to predict the traffic time by google support and machine learning instead of giving it a certain time. Because time and resources are limited, we can’t give the best way to implement this but only the simple one. In the future, with more time and better algorithms, using machine learning with the traffic time data taken from google will give a more accurate time cost.
* Using a tree as input data rather than a matrix for less complicated and reduces the unnecessary loop. With the linked list data structure, the adjacent city with the leftmost\_city and right\_city (not necessarily to be in order), also includes the previous city that we travel from. The loop will be prevented by adding the visited list. The space cost for data input will be less than the matrix and more efficient.

**Maybe one more**

**7. List of bibliographic references**

[**https://docs.goong.io/rest/**](https://docs.goong.io/rest/) **using: API key, distance matrix**