PROJECT REPORT

Subject: Route planning

By

Ta Viet Cuong – 20194422

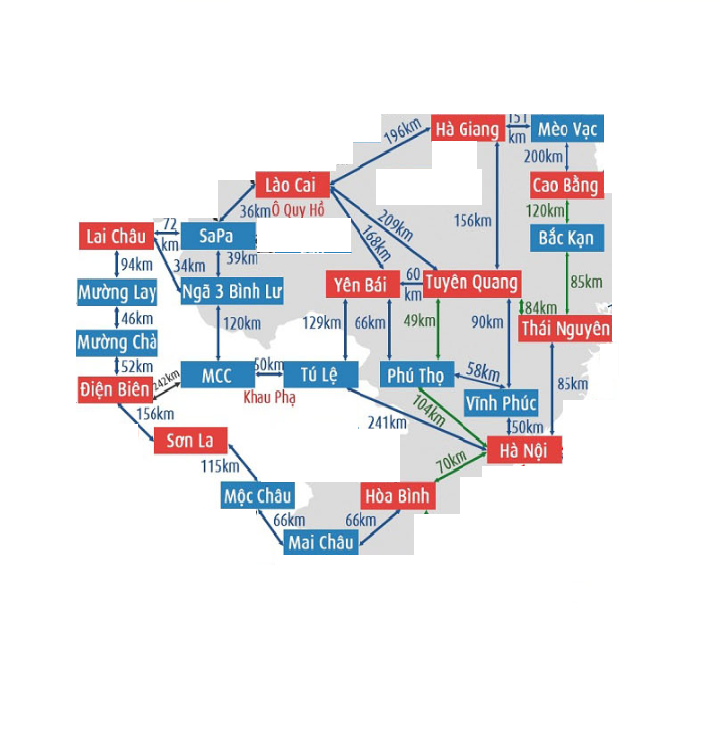
Bui Anh Duc – 20194426

Le Thanh Thang – 20194451

Nguyen Van Toan – 20194457

**1. Presentation of the subject**

Route planning is the method used to find the most effective route about cost, time, … when we move from a place to another.



**2. Description of the problem**

- Write a program to find the cheapest route between two Vietnamese cities and time taken is no more than time allowed. The intelligent vehicle can only travel between 2 adjacent cities, and the objective is to minimize the cost between two cities and satifies time allowed.

- Problem formulation:

+ Initial state: HaNoi

+ Actions model: Action (In: HaNoi) = [Go: HoaBinh, Go: HoaBinh, Go: TuLe, Go: PhuTho, Go: ThaiNguyen]

+ Goal test: LaiChau

+ Path cost: sum of cost, time taken

**3. Selecting the algorithms to be used for solving the problem**

We chose to apply uniform-cost search and A\* search algorithms because:

+ Uniform-cost search and A\* search algorithms are complete and give the optimal solution.

+ Breadth-search algorithm takes much time, exponencial complexity in the worst case.

+ Depth-search and greedy algorithms aren’t complete and don’t give the optimal solution.

**4. Implementing the algorithms to be used for solving the problem**

The main difficulties we had to face for implementation:

+ Making the data took much time, we used city graph in the Internet to make the data.

+ Initially we got stuck in local minima and plateau, such as PhuTho -> TuyenQuang -> PhuTho, then we used CheckInPriority function: when considering PhuTho, we put it in Closed(PriorityQueue) to eliminate it when considering subcities of TuyenQuang then.

**5. Comparing the results of the algorithms used for solving the problem**

a. Providing quantitative performance indicators

|  |  |  |
| --- | --- | --- |
|  | Uniform-cost search | A\* search |
| Percent of the algorithm successfully solved the problem | 100% | 100% |
| Average time complexity (s)  Data 1  Data 2  Data 3  Data 4 | 0.0019  0.00099  0.001  0.0013 | 0.00099  0.00098  0.0099  0.001 |

b. Explaining these results

- Uniform-cost search:

+ Time complexity: Let C\* is Cost of the optimal solution, and ε is each step to get closer to the goal node. Then the number of steps is = C\*/ε+1. Here we have taken +1, as we start from state 0 and end to C\*/ε. Hence, the worst-case time complexity of Uniform-cost search is O(b1 + [C\*/ε])/.

+ Space complexity: The same logic is for space complexity so, the worst-case space complexity of Uniform-cost search is O(b1 + [C\*/ε]).

- A\* search:

+ Time complexity: The number of nodes expanded is still exponential in the length of the solution (path)

+ Space complexity: It keeps all generated nodes in memory. Hence space is the major problem here, not time

**6. Conclusion and possible extensions**

- Uniform-cost search algorithm:

+ The first solution found is also the one has the lowest cost.

+ If the problem has a solution, the algorithm will stop.

+ If all branches have equal cost, the algorithm becomes breadth-search algorithm.

- A\* search algorithm:

+ The first solution found is also the one has the lowest cost.

**7. List of tasks**

- Programming tasks:

+ Implementing uniform-cost search algorithm: Nguyen Van Toan 70%, Le Thanh Thang 30%

+ Implementing A\* search algorithm: Bui Anh Duc 70%, Ta Viet Cuong 30%

- Analytic tasks:

+ Writing the report: Ta Viet Cuong

+ Writing the presentation: Nguyen Van Toan 50%, Le Thanh Thang 50%

+ Creating the demo video: Bui Anh Duc

**8. List of bibliographic references**

<https://en.wikipedia.org/wiki/Priority_queue?fbclid=IwAR3rqZo7v3e2mrD61hPnmqafAQujwFyYIpG1tQPHB2-8wYF5f_M4ao4ALrY>

<https://docs.python.org/3/library/collections.html?fbclid=IwAR2I5RkRBoGqb61MjeT2qTbr9jyW0Q4Vy1WZR8O5psQoPmmCgaUDeV_kV80>

[https://en.wikipedia.org/wiki/A\*\_search\_algorithm?fbclid=IwAR2prKF8IH8uVfFp8GqF9QyZkl04xkWhcFhsbe5imsHfsaE30rlHO9aOdlI#:~:text=recently%20expanded%20node.-,Complexity,number%20of%20successors%20per%20state](https://en.wikipedia.org/wiki/A*_search_algorithm?fbclid=IwAR2prKF8IH8uVfFp8GqF9QyZkl04xkWhcFhsbe5imsHfsaE30rlHO9aOdlI#:~:text=recently%20expanded%20node.-,Complexity,number%20of%20successors%20per%20state)

<https://vi.wikipedia.org/wiki/T%C3%ACm_ki%E1%BA%BFm_chi_ph%C3%AD_%C4%91%E1%BB%81u?fbclid=IwAR3sbET_pW0hHAb87LD6zCPkn2CdyXDWmAsXGACrIyki62B_Fjtuwk86XUk#Thu%E1%BA%ADt_to%C3%A1n>