

# Assignment 1, part 2

## Part 2:

Explanation of the problem (1.5 points)

Given a list of cities with link distance values between adjacent cities, find a path from starting city to the target city. Greedy Best-first search will return the first solution found and A\* will return the optimum path.

a. Give the representation of a solution (answer) of the problem, as explained during the course. (0.5)

Given:

$p$  - number of cities

vector  $C = \{c_1, c_2, c_3 \dots c_p\}$  representing the names of  $n$  cities

Matrix  $D$ , where  $D[i][j]$  represents path cost between city  $i$  and city  $j$ ,

$0 \leq i, j \leq p$

Matrix  $H$ , where  $H[i][j]$  represents straight line distance between city  $i$  and city  $j$ ,

$0 \leq i, j \leq p$

int start, int end - indexes of the start and goal cities in  $C$

The solution will be a vector  $V = \{\text{start}, \dots, \text{end}\}$  of length  $n$ , where  $V[i]$  = index of the city visited at step  $i$  from the vector  $C$ .

b. Give the equation of  $f(n)$  used in Greedy Best-first Search (or Explain how to calculate  $f(n)$ ). (0.25)

$$f(n) = h(n)$$

With  $h(n)$  representing heuristic information function (straight line distance between city  $n$  and goal city)

c. Give the equation of  $f(n)$  used in A\* (or Explain how to calculate  $f(n)$ ). (0.25)

$$f(n) = g(n) + h(n)$$

Where  $h(n)$  is same as above and  $g(n)$  represents the cost to reach node  $n$  from the start node

In my implementation, I compare the value  $D[\text{current}][n] + H[n]$  to determine the next node on the path.

d. Explain both algorithms and the differences between them. (0.5)

Greedy Best-first Search in my implementation is an algorithm that greedily chooses the next city on a path based on the shortest straight line distance to the goal city from a candidate city. It first finds a path from the start node to city 2, and then recurses until it reaches target city. This algorithm returns the first solution path that it comes across.

A\* search in my implementation is an algorithm that chooses the next city which has the minimum combined value of the heuristic distance from the candidate node to the goal node and the path cost between current node and the candidate node. It starts with the start node, finds optimal 2d node and so on until after the target is reached.

The difference is that Greedy algorithm only considers the heuristic values, which do not always yield the optimal solution, and in practice while running my code I always got a shorter path from the A\* approach, since it ensures optimality via consideration of link costs as well.