Introduction to Artificial Intelligence - LAB 1

ISEP –November 20, 2024

Instructions: Prepare an **individual** report including the **source code in Python** and the **numerical results**. Submit the report on Teams. Please respect the deadline of **November 27th** to submit your report. Any delay will result in penalties. The use of existing modules on the internet is permitted provided that you cite the **bibliographic resources.**

Introduction

In this laboratory, we will analyze the use of the A* search algorithm. The work should be deposited on Teams.

The A* search algorithm

The A* algorithm is basically an artificial intelligence problem used for path finding (from point A to point B) and graph traversals.

This algorithm is the advanced form of the BFS (breadth-first search) algorithm, which searches for the shortest path first than the longest paths. It is a complete and optimal solution to solve path and tree problems. This algorithm can be used in a wide range of contexts. The A* search algorithm uses the cost of the heuristic path, the cost of the starting point, and the ending point.

As a reminder, this algorithm uses the calculation of the following costs:

1. $f(n) = g(n) + h(n)$.
2. $g(n)$: the actual cost path from the start node to the current node.
3. $h(n)$: the actual cost path from the current node to the target node.
4. $f(n)$: the actual cost path from the start node to the end node.

Question 1. Definitions and properties

1.1 Give the definition of the optimality of an algorithm;

1.2 Give the definition of the completeness of an algorithm;

1.3 Give the Difference between the A* algorithm and the Uniform-cost Search algorithm;

Question 2. Graph Search

Write the algorithm (pseudo-code) for searching in graphs (Graph Search). You will need for the implementation of the A* algorithm;

Question 3. Implementation of A* research method

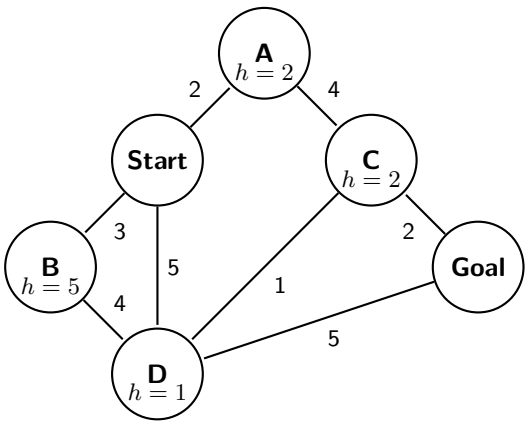
Part A.

For the implementation of the A* algorithm, we need to use two arrays, namely FRINGE and CLOSED. FRINGE represents an array that contains nodes that have been spawned but have not been examined so far. CLOSED represents an array that contains the nodes that are being examined. To implement the A* algorithm, you can follow the following steps.

1. First, place the starting node in FRINGE and find its value $f(n)$ by calculating $f(n) = g(n) + h(n)$
2. Then remove the node from FRINGE, having the smallest value $f(n)$. If it is a Goal node, stop and return SUCCESS.
3. Otherwise, remove the node from the FRINGE list and search for all its successors using a successor() function for example.
4. Find the value $f(n)$ of all successors, put them in the FRINGE list, and put the deleted node in CLOSED.
5. Go to step 2.
6. Return the solution

Part B.

1. Apply the A* algorithm on the following graph :



2. Display the parsed nodes as well as the final result.