Homework Artificial Intelligence

Domenico Alfano

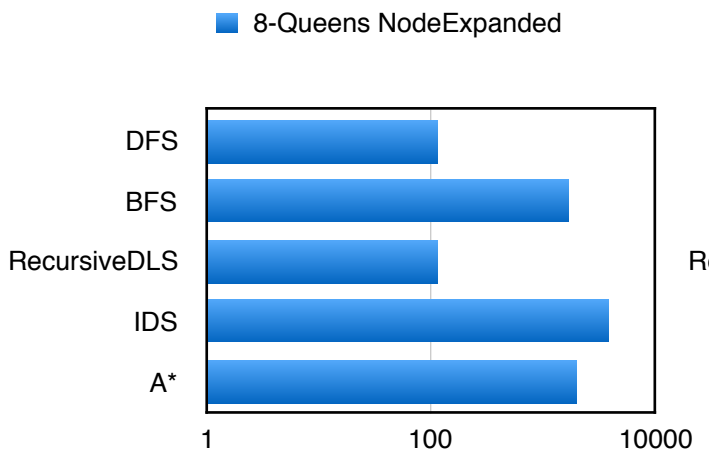
1752007

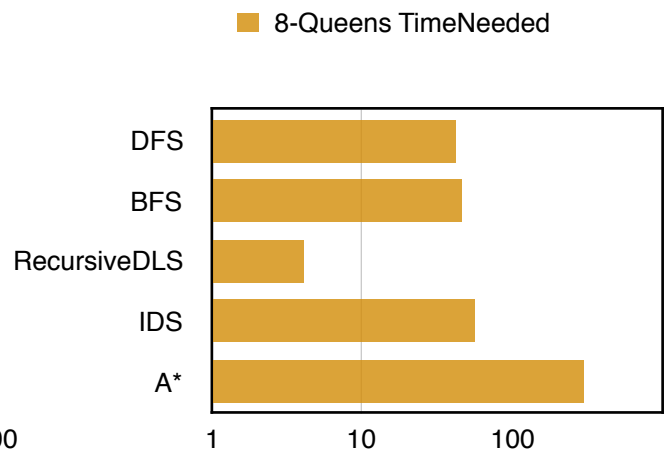
Part 1.1

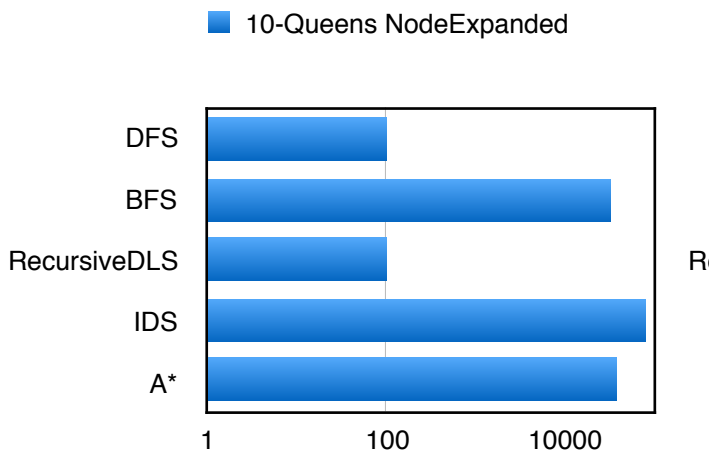
N-Queens Problem Analysis

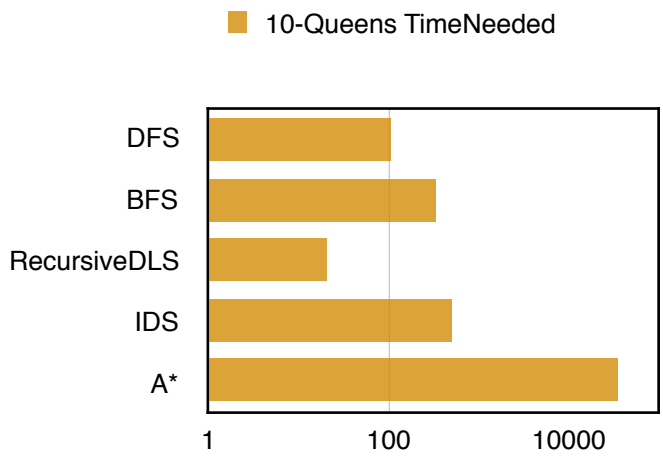
In the first point of HomeWork1, I have analyzed the N-queens problem, with N= 8;10,12. In particular I've plot, with a logarithmic scale, the number of nodes generated and the time needed (in milliseconds) to find the solution by the search algorithm.

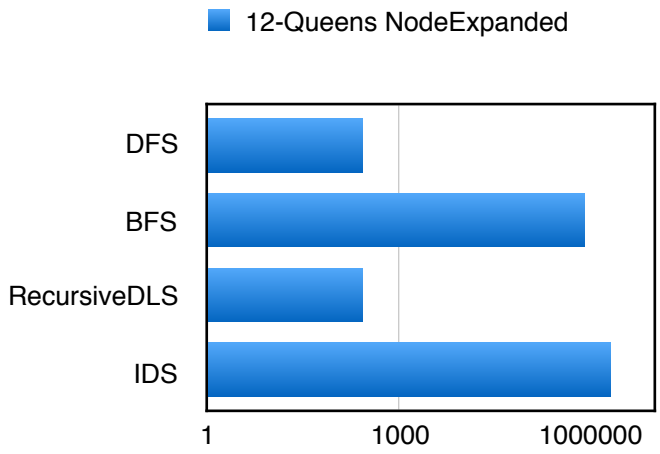
The algorithms without an admissible solution will not appear in the plot (for clarity reason).

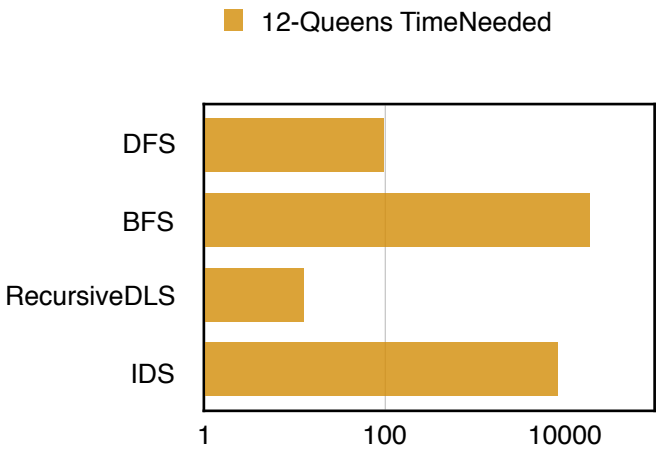








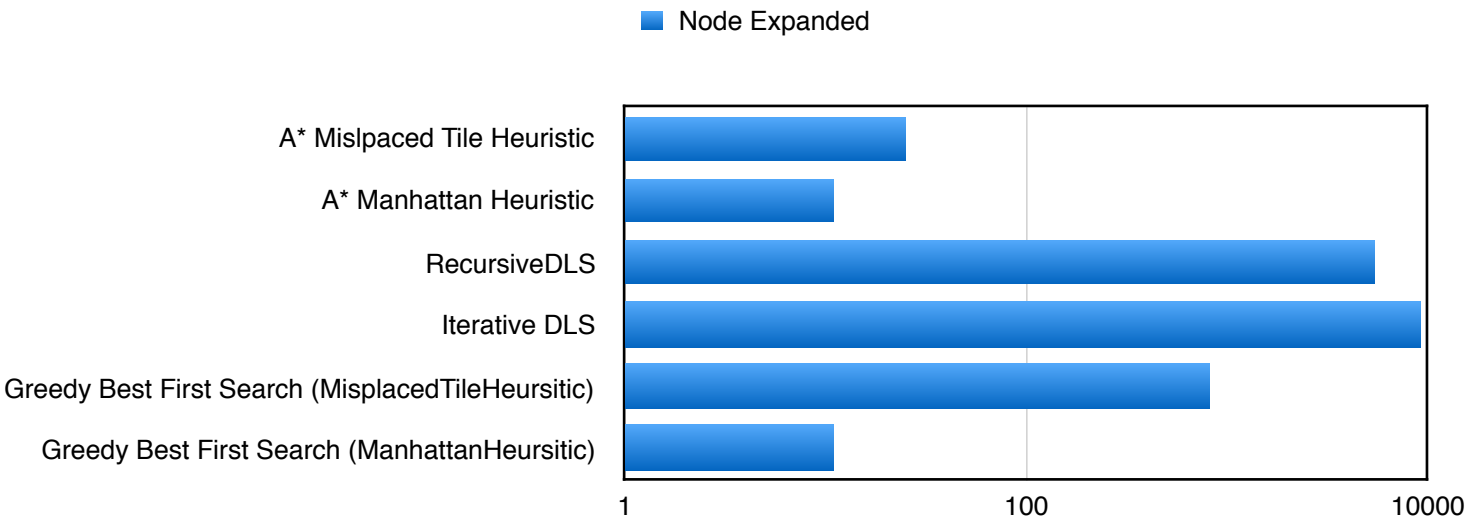

12-Queens NodeExpanded
DFS
BFS
RecursiveDLS
IDS
1
1000
1000000

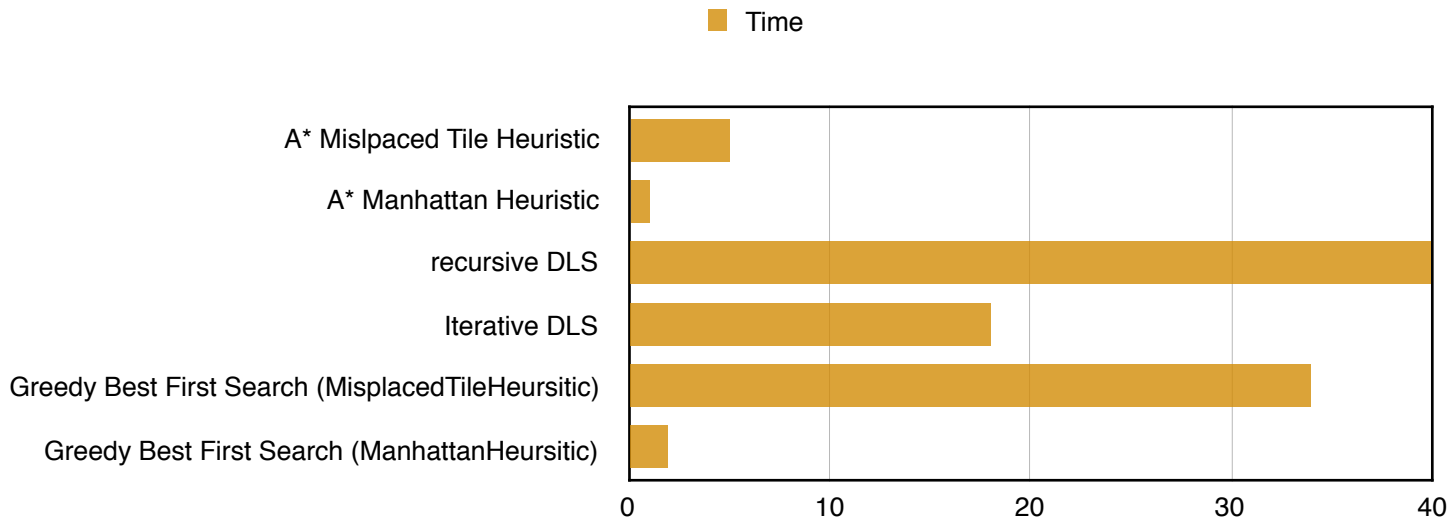

12-Queens TimeNeeded
DFS
BFS
RecursiveDLS
IDS
1
100
10000

Part 1.2

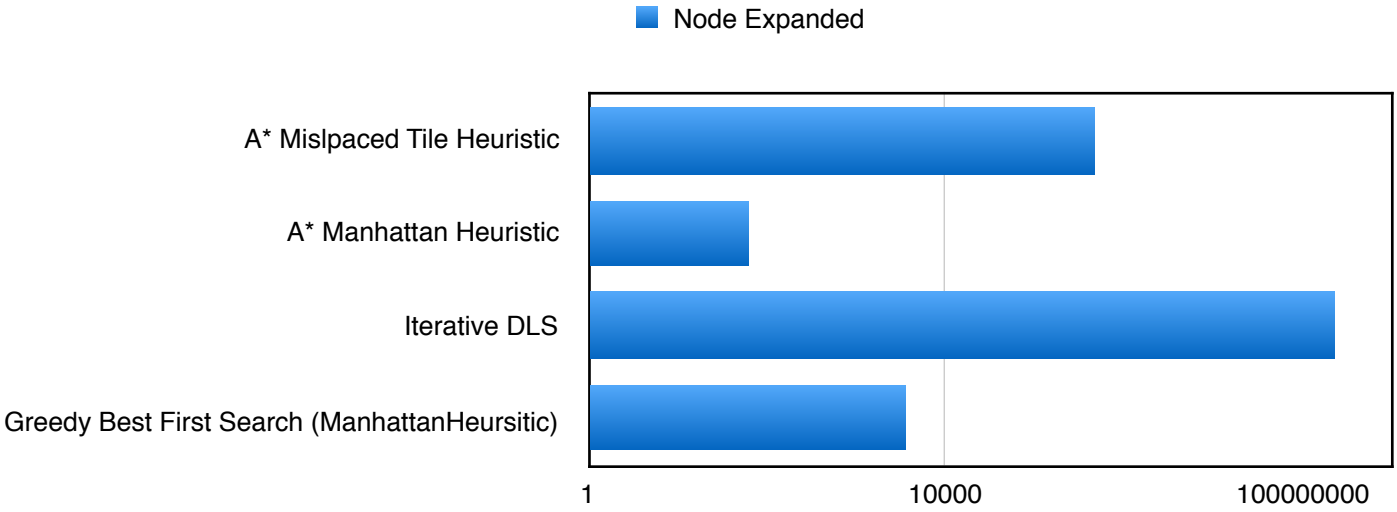
8-Puzzle Problem Analysis

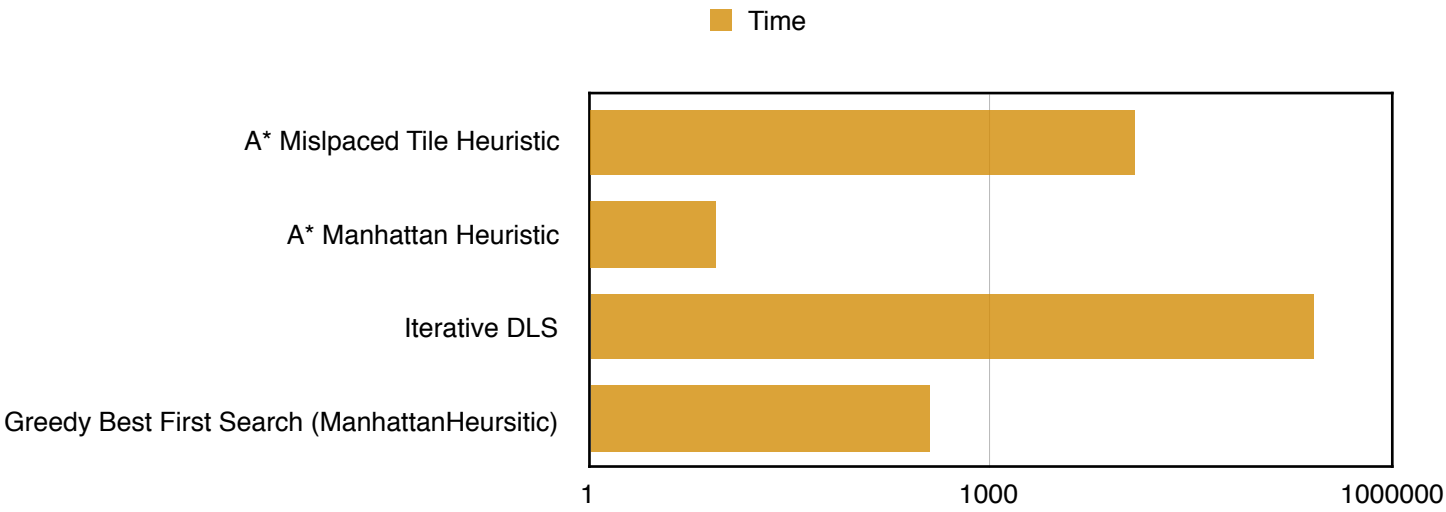
In the second point of the Homework, I have analyzed the 8-puzzle problem two times. In the first one I familiarized with the configuration of the board and I tried to solve the game with all the algorithms which are in the EightPuzzleApp.java. For each run was been plotted the number of nodes generated and the time needed to find the solution with the Medium configuration.





The second approach to this game has previewed the extension of the state space by creating a 4 × 4 board and the creation of a new Medium start configuration to repeat the previous procedure. The algorithms without an admissible solution will not appear in the plot.





Part 2

Domestic Robot Problem

In this part of the homework we have to define a workspace for moving our robot from an initial position to a goal position. For doing this we can use the aima framework and so we have to make the principal component of it , that are the following :

State : is the robot position .

Initial State : is the start position of the robot.

Action : is the action of the robot that can do in that position.

Transition Model : is the execution of an action in that position.

Goal : is the goal end position.

Code Modify :

At the begin I add **position** to mine Environment , so I can describe where the robot is .

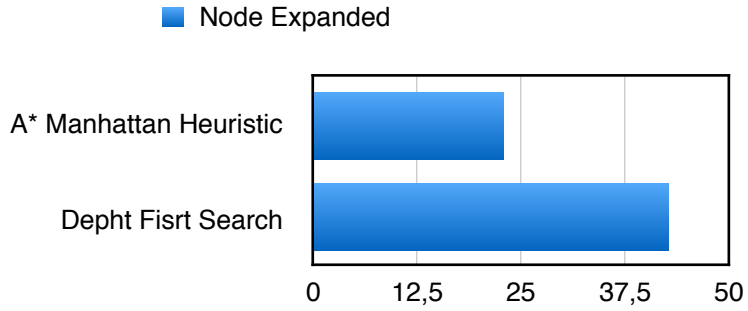
Then I create the class **Action** that describe what action can take place when a position is given as argument to this fuction. It return what are the possible moves in a range of all the four possible action {Right,Left,Top,Down}.

After I create the class **Result** that given a position and an action it execute that action.

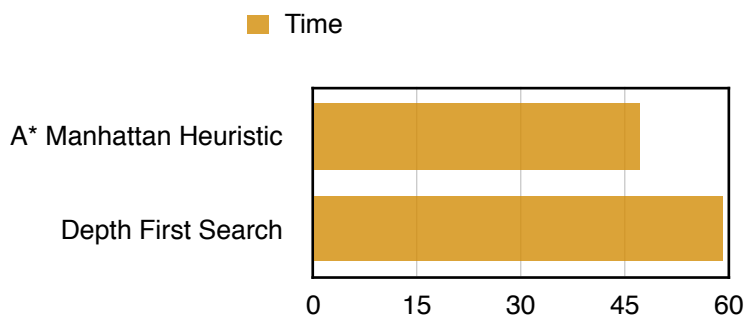
Then I create a class **Goal** that check if the position of the robot is the goal position and return true or false from this assumption.

At the end I create a Class for the Manhattan **heuristic** of the A* Search for describe the minimal distance from the start position and the goal position.

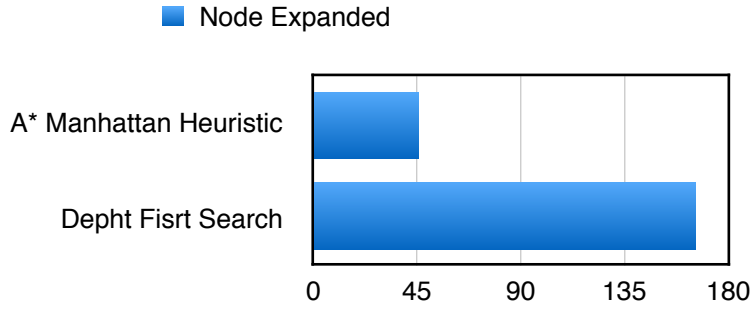
The Depth First algorithm need only the creation of the problem solver so we can easy use it . Here the result plotted with the blue path for the best path find by the two search algorithm with the of the number of node expanded and the time occurred .



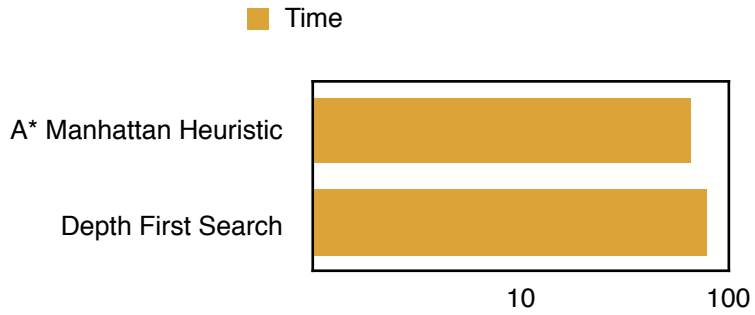
A* Manhattan 16x16



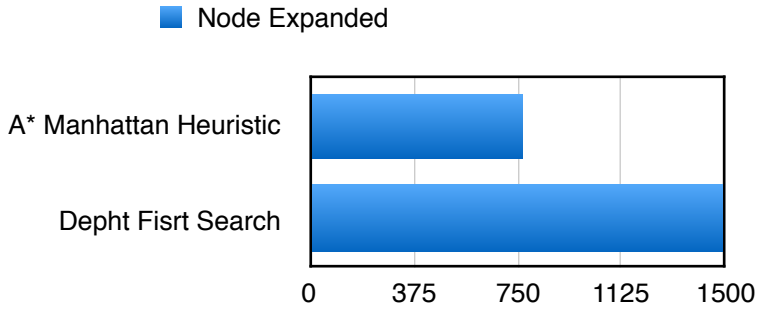
Depth First 16x16



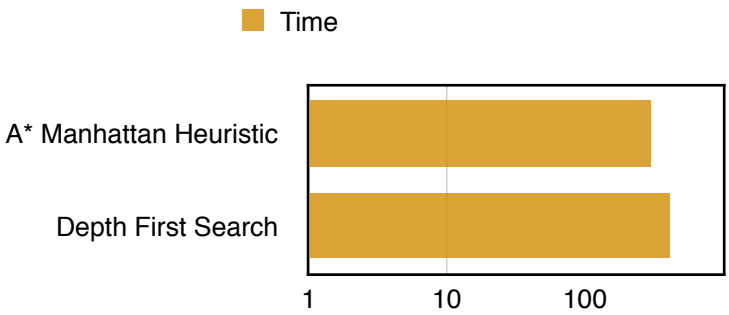
A* Manhattan 32x32

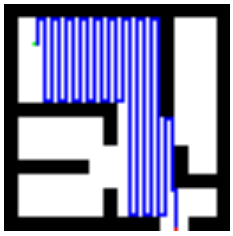


Depth First 32x32



A* Manhattan 32x32





Depth First 32x32