University of Barcelona

Facultat de Matemàtiques i Informàtica

Report Project 1: Elementary Search Algorithms

Artificial Intelligence

Junjie Li and Manuel Liu Wang

October 2022

In order to implement the three algorithms, we implemented the isCheckMate method, this method checks all the checkmate positions around the black king and returns true or false. For all the three implementations, we use a method called getPath, this method gets the correct path towards the checkmate state.

1 Depth First Search(DFS)

For the DFS implementation, it starts with a node passed on via parameter, the root, called currentState, and an int that acts like a limiter, called depth. With the root, we start exploring the graph in depth, checking every possible move in a recursive mode. In every recursive call, we pass the current state and sum 1 to the current depth.

The limiter depth checks if the number of steps made by the algorithm is superior to our maxDepth, if that is the case, it ends the exploration in that branch and continues with the next one, since we don't want a path that has more steps than the value in maxDepth.

Once the algorithm finds a path to one of the checkmates states, it returns that path, it doesn't search for the most optimized one, instead it searches for the first path to it.

2 Breadth First Search(BFS)

In the BFS implementation, our start is the same as in the DFS, except the part that it only has the root passed as parameter. The BFS explores all nodes in the same depth before moving to the next level or getting into one of the checkmate state, for this algorithm we use a Queue in a FIFO (First In First Out) mode.

When it finished exploring all the nodes, it returns the path to the checkmate position through the getPath method.

3 A* Search

For the AStar, we use a method to help us calculate heuristic values, it returns the cost/distance towards the target state from the state passed via parameter. The steps we take on this algorithm are the ones that cost less, for that, we compare the value that we have with the new one, the new value comes from the next step we want to take plus the distance to the target from this step, keep the lowest one between these two values.

4 Do your algorithms work in precisely the same way? Did you have to change anything?

Yes, they work in the same way with the king in [7][4] and in [7][7]. In the BFS implementation, it goes a bit faster in [7][7] than in [7][4]. I didn't have to change anything since the algorithms works with the current state passed via parameter.