 Campus Querétaro

Cristian Alexsis González Jaime A01204478

Luis Pablo Morales Cruz A01205724

Jesús Danniel Milacatl Avila A01204880

***Implementing Uninformed search Algorithms***

Teacher: Ruben Stranders

**Report**

Test your program with a couple of different problems. Increase the size of the problem to test the limits of your program.

Make a table comparing how **many nodes are searched** to find the answer for each problem. For this table, you should compare a number of different problems (at least 3) to avoid a statistical bias.

Depth First Search

Breadth First Search

A\*

1. Which heuristics did you use for the A\* algorithm?

For our A\* search we use an heuristics based in the position of the blocks, if the blocks aren’t out of their position final , beside of normal cost for each move , the heuristic has another cost for each move that is needed to put the block in the correct place

* Normal cost
  + Picking up the container takes 0.5
  + Moving the container one stack to the left or right takes 1
  + Putting the container down takes 0.5
* Heuristic
  + Moving the container one stack 1

B B

A C D 🡪 A C D

Initial State Goal State

In this case the only block that isn’t in its place, is the block B, for that reason the only movement that is needed is to move this block from the stack 0 to the stack 2 ,B(0,2), we don’t need to move another block neither expand others nodes to search more states, because the blocks A,C and D are in their correct place.  
So the Cost for the heuristic in this case is equal to 2 more the normal cost for each movement, the final cost will be equal to 5

1. Which of the four algorithms searches the least nodes and which one take the most?

The algorithms with the most number of nodes expanded was the Depth Search, because it needs to expand all the nodes to found the most efficient solution and the search with the least number of nodes expanded was the A\* , thanks to the heuristic implemented in the algorithms, it found the solution with a less number of nodes expanded

1. Why does this happen?

Because the algorithm for Depth search is based in expand all the possible nodes to find the shallowest solution, the only problem is that while bigger is the problem, more memory is needed. On the other hand A\* Search, is the most efficient algorithm if the heuristic is implemented correctly, because with a good heuristic is possible to find the shallowest solution expanding the least number of nodes possible.

1. Which algorithms are optimal? Why?

Depends the mean of optimal, for example if is looking for the shallowest solution, the best algorithm in this case is the Depth Search, but also it is which need more memory and more time to find a solution besides sometimes the memory isn’t enough But if is looking for an efficient solution without a lot of memory, the best algorithm is A\*, the problem is that it need a good heuristic although it won’t give us the shallowest solution

1. In your opinion what are the benefits of simpler algorithms versus more complex ones?

It help us to find a solution faster that the complex ones, but for a good implementation is needed to find a good system of heuristic and cost. Although not always they find the shallowest solution like a complex, but nevertheless they don’t need too much memory to solve the problem, maybe in a future with more powerful computers, we can use the complex algorithms to find the shallowest solution in the biggest problems.