Arturo Alam Téllez Villagómez A01205569
Fermin Alejandro Moreno Cornejo A01209665

Your report needs to explain at least the following:

- Which heuristics did you use for the A* algorithm?
    - We used the heuristic of determining the amount of boxes that are not in their place
    - For the inconsistent we used a random value for each node that can be either good or bad
- 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. Which of the three algorithms (UCS, A *with consistent and and A* with an inconsistent heuristic) searches the least nodes and which one take the most?

Max	Initial State	Goal State	UCS visited	A* consistent visited	A* inconsistent visited
4	(A, B); (C, D); ()	(A, B, C); X; X	17	10	6
3	(); (A, C); (B)	(B); (A); (C)	16	8	66
7	(A, C, D); (T, Y); ()	(C); (D, A); X	9049	3422	13161

- Why does this happen?
    - Because UCS is an optimal solution but it checks all possible paths to determine which one is better, when you give a consistent heuristic it is supposed to be better since you are helping the algorithm to make decisions without the need to do all the computations.
- Which algorithms are optimal? Why?
    - UCS is optimal because it does all the comparisons and looks always for the optimal result, but having a heuristic it is supposed to make things faster, but if the heuristic is badly designed the algorithm won't be optimal.
- In your opinion, what are the benefits of simpler algorithms versus more complex ones?

- Simpler algorithms are easier to understand and to program, more complex ones require more time to think of for example in a good heuristic that makes the program better and require also to think on limitations of the tools used.