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.

Your report needs to explain at least the following:

1. Which heuristics did you use for the A* algorithm?
    a. The minimum number of moves needed for a specific crate to be in the right stack. H(nodeCurrentStack, finalState) = abs (nodeCurrentStack - stackItShouldBe)

2. Which of the four algorithms searches the least nodes and which one take the most?
    a. Least Nodes: A* Consistent
    b. Most Nodes: DF

3. Why does this happen?
    a. Because the Depth first explores all of the possible branches without even if it loops or if it shouldn't move a crate. Without taking into consideration any costs of heuristics

4. Which algorithms are optimal? Why?
    a. For this type of search graphs where there's an available heuristic and a path cost and states don't change overtime nor can be repeated, this is optimal.
    b. BFS could also be considered complete, though not optimal

5. In your opinion what are the benefits of simpler algorithms versus more complex ones?
    a. If you don't have a possible heuristic nor path cost then A* is as optimal as a greedy best first search

TESTCASE	BFS	DFS	A* non consistent	A* consistent
3 (B, A); (C, D, E); () (D, C); X; (B, A, E)	25	1194	23	23
4 (B, A); (C, D, E); (F,G) (D, C); (G,F); (B, A, E)	39	12860	37	34
3 (B, A); (C, D, E); (E, C); (D, A,B);	21	2486	21	21