

Alexandre Labbe, alabbe, 2377032

Part 1 Step 4 (c)

Checks that if the given move leaves any humans, that the humans will not be outnumbered.

If humans will be left on that side, if there will be more robots, return false.

Part 1 Step 8

	DFS	BFS
Farmer Fox	Path found: Cross with chicken, Cross alone, Cross with fox, Cross with chicken, Cross with grain, Cross alone, Cross with chicken. Length of path: 7 edges Number of states expanded: 7	Path found: Cross with chicken, Cross alone, Cross with fox, Cross with chicken, Cross with grain, Cross alone, Cross with chicken. Length of path: 7 edges Number of states expanded: 19
Human Robots	Path found: [1,1], [1,0], [3,0], [2,0], [1,1], [2,0], [3,0], [1,0], [1,1] Length of path: 9 edges Number of states expanded: 10	Path found: [1,1], [1,0], [3,0], [2,0], [2,1], [1,0], [1,1] Length of path: 7 edges Number of states expanded: 10
Hanoi 4 Rings	Path found: [[4, 3, 2, 1],[],[]] [[4, 3, 2],[1],[]] [[4, 3],[1],[2]] [[4, 3, 1],[],[2]] [[4, 3],[],[2, 1]] [[4],[3],[2, 1]] [[4, 1],[3],[2]] [[4],[3, 1],[2]] [[4, 2],[3, 1],[]] [[4, 2, 1],[3],[]] [[4, 2],[3],[1]] [[4],[3, 2],[1]] [[4, 1],[3, 2],[]] [[4],[3, 2, 1],[]] [[],[3, 2, 1],[4]] [[1],[3, 2],[4]] [[],[3, 2],[4, 1]] [[2],[3],[4, 1]] [[2, 1],[3],[4]] [[2],[3, 1],[4]] [[],[3, 1],[4, 2]]	Path found: [[4, 3, 2, 1],[],[[]] [[4, 3, 2],[1],[[]] [[4, 3],[1],[2]] [[4, 3],[],[2, 1]] [[4],[3],[2, 1]] [[4, 1],[3],[2]] [[4, 1],[3, 2],[[]] [[4],[3, 2, 1],[[]] [[],[3, 2, 1],[4]] [[],[3, 2],[4, 1]] [[2],[3],[4, 1]] [[2, 1],[3],[4]] [[2, 1],[],[4, 3]] [[2],[1],[4, 3]] [[],[1],[4, 3, 2]] [[],[],[4, 3, 2, 1]] Length of path: 15 edges Number of states expanded: 70

	[[1],[3],[4,2]] [[],[3],[4,2,1]] [[3],[],[4,2,1]] [[3,1],[],[4,2]] [[3],[1],[4,2]] [[3,2],[1],[4]] [[3,2,1],[],[4]] [[3,2],[],[4,1]] [[3],[2],[4,1]] [[3,1],[2],[4]] [[3],[2,1],[4]] [[],[2,1],[4,3]] [[1],[2],[4,3]] [[],[2],[4,3,1]] [[2],[],[4,3,1]] [[2,1],[],[4,3]] [[2],[1],[4,3]] [[],[1],[4,3,2]] [[1],[],[4,3,2]] [[],[],[4,3,2,1]] Length of path: 40 edges Number of states expanded: 40	
--	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--

Towers-of-Hanoi BFS and DFS algorithm analysis.

The maximum length of the open list is greater when using breadth first search. This is because in the case of BFS, the size of the branching factor greatly impacts the size of the open list. Towers-of-Hanoi has a branching factor between 2 and 3, since there are normally 2 or 3 different operators you can take from a given state. Since there is consistently a large branching factor, more states will end up on the open list.

The solution path length is shorter when using breadth first search. This is because breadth first search will not go deeper into the state space graph, therefore it will find the shortest path to the solution possible but will possibly use a lot of memory. Depth first search will go deep into the graph to try and find a solution, which will possibly result in a long solution, but less memory may have had to be used in the process.