



















Assignment 3: BlocksWorld


Fiona O'Connell, Nathan Hafely

Group 14 Contributions: All contributed equally.

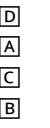

Results:

	TEST 1: (4 blocks)
<u>Start</u>	<p>start([[on,a,b],[on,b,table],[on,c,d],[clear,a],[clear,c],[on,d,table]]).</p> <div> <div>A</div> <div>C</div> <div>B</div> <div>D</div> </div>
<u>Goal</u>	<p>goal([[on,d,a],[on,a,c],[on,c,b],[on,b,table],[clear,d]]).</p> <div> <div>D</div> <div>A</div> <div>C</div> <div>B</div> </div>
<u>Steps (DFS)</u>	<p>[[[on,a,b],[on,b,table],[on,c,d],[clear,a],[clear,c],[on,d,table]], [[on,a,c],[on,b,table],[on,c,d],[clear,a],[clear,b],[on,d,table]], [[clear,c],[on,a,table],[on,b,table],[on,c,d],[clear,a],[clear,b],[on,d,table]], [[clear,c],[on,a,table],[on,b,table],[on,c,a],[clear,d],[clear,b],[on,d,table]], [[clear,c],[on,a,table],[on,b,table],[on,c,b],[clear,d],[clear,a],[on,d,table]], [[clear,b],[clear,c],[on,a,table],[on,b,table],[on,c,table],[clear,d],[clear,a],[on,d,table]], [[clear,b],[clear,c],[on,a,table],[on,b,a],[on,c,table],[clear,d],[on,d,table]], [[clear,b],[clear,a],[on,a,table],[on,b,c],[on,c,table],[clear,d],[on,d,table]], [[clear,b],[clear,a],[on,a,table],[on,b,d],[on,c,table],[clear,c],[on,d,table]], [[clear,a],[on,a,b],[on,b,d],[on,c,table],[clear,c],[on,d,table]], [[clear,a],[on,a,c],[on,b,d],[on,c,table],[clear,b],[on,d,table]], [[clear,d],[on,a,c],[on,b,a],[on,c,table],[clear,b],[on,d,table]], [[clear,a],[clear,d],[on,a,c],[on,b,table],[on,c,table],[clear,b],[on,d,table]], [[clear,a],[clear,c],[on,a,d],[on,b,table],[on,c,table],[clear,b],[on,d,table]], [[clear,c],[on,a,d],[on,b,table],[on,c,a],[clear,b],[on,d,table]], [[clear,c],[on,a,d],[on,b,table],[on,c,b],[clear,a],[on,d,table]], [[clear,d],[on,a,c],[on,b,table],[on,c,b],[clear,a],[on,d,table]], [[clear,d],[on,a,c],[on,b,table],[on,c,b],[on,d,a]]]</p> <p>Path Length: 18</p>

	state[0]	state[1]	state[2]	state[3]	state[4]	state[5]	state[6]
							
	state[7]	state[8]	state[9]	state[10]	state[11]	state[12]	state[13]
							
	state[14]	state[15]	state[16]	state[17]			
							

Notes	<p>The program successfully found a solution path, but explored far more states than necessary to reach the goal, since it's an uninformed DFS.</p> 
--------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Updated Notes	<p>Test 4 couldn't find a solution, so I implemented Iterative Deepening Depth First Search (IDDFS).</p> <p>Revised Path: (Length: 5)</p> <pre>[[on,a,b],[on,b,table],[on,c,d],[clear,a],[clear,c],[on,d,table]], [[clear,b],[on,a,table],[on,b,table],[on,c,d],[clear,a],[clear,c],[on,d,table]], [[clear,d],[on,a,table],[on,b,table],[on,c,b],[clear,a],[clear,c],[on,d,table]], [[clear,d],[on,a,c],[on,b,table],[on,c,b],[clear,a],[on,d,table]], [[clear,d],[on,a,c],[on,b,table],[on,c,b],[on,d,a]],</pre>
----------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

	TEST 2: (4 blocks)
Start	<p>start([[on,a,c],[on,b,table],[on,c,b],[clear,d],[on,d,a]]).</p> 
Goal	<p>goal([[on,b,c],[on,a,table],[on,c,table],[on,d,table],[clear,b],[clear,a],[clear,d]]).</p> 
Steps (DFS)	<pre>[[[on,a,c],[on,b,table],[on,c,b],[clear,d],[on,d,a]], [[clear,a],[on,a,c],[on,b,table],[on,c,b],[clear,d],[on,d,table]], [[clear,a],[on,a,d],[on,b,table],[on,c,b],[clear,c],[on,d,table]],</pre>

	<div>[[clear,b],[on,a,d],[on,b,table],[on,c,a],[clear,c],[on,d,table]], [[clear,a],[clear,b],[on,a,d],[on,b,table],[on,c,table],[clear,c],[on,d,table]], [[clear,a],[clear,d],[on,a,b],[on,b,table],[on,c,table],[clear,c],[on,d,table]], [[clear,a],[clear,d],[on,a,c],[on,b,table],[on,c,table],[clear,b],[on,d,table]], [[clear,c],[clear,a],[clear,d],[on,a,table],[on,b,table],[on,c,table],[clear,b],[on,d,table]], [[clear,c],[clear,d],[on,a,table],[on,b,table],[on,c,a],[clear,b],[on,d,table]], [[clear,c],[clear,a],[on,a,table],[on,b,table],[on,c,d],[clear,b],[on,d,table]], [[clear,c],[clear,a],[on,a,table],[on,b,table],[on,c,b],[clear,d],[on,d,table]], [[clear,c],[on,a,table],[on,b,table],[on,c,b],[clear,d],[on,d,a]], [[clear,c],[on,a,table],[on,b,table],[on,c,d],[clear,b],[on,d,a]], [[clear,d],[clear,c],[on,a,table],[on,b,table],[on,c,table],[clear,b],[on,d,a]], [[clear,d],[clear,a],[on,a,table],[on,b,table],[on,c,table],[clear,b],[on,d,c]], [[clear,d],[on,a,table],[on,b,a],[on,c,table],[clear,b],[on,d,c]], [[clear,d],[on,a,table],[on,b,a],[on,c,table],[clear,c],[on,d,b]], [[clear,b],[clear,d],[on,a,table],[on,b,a],[on,c,table],[clear,c],[on,d,table]], [[clear,b],[clear,a],[on,a,table],[on,b,d],[on,c,table],[clear,c],[on,d,table]], [[clear,b],[clear,a],[on,a,table],[on,b,c],[on,c,table],[clear,d],[on,d,table]]]</div> <div>Path Length: 20</div>																					
	<table><tr><td>state[0] D A C B</td><td>state[1] A C B D</td><td>state[2] C A B D</td><td>state[3] C A B D</td><td>state[4] A B C D</td><td>state[5] A B C D</td><td>state[6] A B C D</td></tr><tr><td>state[7] A B C D</td><td>state[8] C A B D</td><td>state[9] C A B D</td><td>state[10] C A B D</td><td>state[11] D C A B</td><td>state[12] C D A B</td><td>state[13] D A B C</td></tr><tr><td>state[14] D A B C</td><td>state[15] B D A C</td><td>state[16] D B A C</td><td>state[17] B A C D</td><td>state[18] B A C D</td><td>state[19] B A C D</td><td></td></tr></table>	state[0] D A C B	state[1] A C B D	state[2] C A B D	state[3] C A B D	state[4] A B C D	state[5] A B C D	state[6] A B C D	state[7] A B C D	state[8] C A B D	state[9] C A B D	state[10] C A B D	state[11] D C A B	state[12] C D A B	state[13] D A B C	state[14] D A B C	state[15] B D A C	state[16] D B A C	state[17] B A C D	state[18] B A C D	state[19] B A C D	
state[0] D A C B	state[1] A C B D	state[2] C A B D	state[3] C A B D	state[4] A B C D	state[5] A B C D	state[6] A B C D																
state[7] A B C D	state[8] C A B D	state[9] C A B D	state[10] C A B D	state[11] D C A B	state[12] C D A B	state[13] D A B C																
state[14] D A B C	state[15] B D A C	state[16] D B A C	state[17] B A C D	state[18] B A C D	state[19] B A C D																	
Notes	Similar to test 1. A solution was found but the returned path was not efficient.																					
Updated Notes	<div>Revised (IDDFS): (Length: 5)</div> <div>[[on,a,c],[on,b,table],[on,c,b],[clear,d],[on,d,a]], [[clear,a],[on,a,c],[on,b,table],[on,c,b],[clear,d],[on,d,table]], [[clear,c],[clear,a],[on,a,table],[on,b,table],[on,c,b],[clear,d],[on,d,table]], [[clear,b],[clear,c],[clear,a],[on,a,table],[on,b,table],[on,c,table],[clear,d],[on,d,table]], [[clear,b],[clear,a],[on,a,table],[on,b,c],[on,c,table],[clear,d],[on,d,table]],</div> <table><tr><td>state[0] D A C B</td><td>state[1] A C B D</td><td>state[2] C A B D</td><td>state[3] A B C D</td><td>state[4] B A C D</td></tr></table>	state[0] D A C B	state[1] A C B D	state[2] C A B D	state[3] A B C D	state[4] B A C D																
state[0] D A C B	state[1] A C B D	state[2] C A B D	state[3] A B C D	state[4] B A C D																		

TEST 4: (6 blocks)

Start	start([on,e,d],[on,d,a],[on,a,table],[on,b,c],[on,c,table],[on,f,table],[clear,e],[clear,b],[clear,f])). <div> <div>E</div> <div>D B</div> <div>A C F</div> </div>
Goal	goal([on,e,a],[on,a,b],[on,b,c],[on,c,table],[on,d,f],[on,f,table],[clear,e],[clear,d])). <div> <div>E</div> <div>A</div> <div>B D</div> <div>C F</div> </div>
Steps (DFS)	[....] [[clear,e],[clear,b],[clear,a],[on,e,d],[on,d,f],[on,a,c],[on,b,table],[on,c,table],[on,f,table]], [[clear,c],[clear,b],[clear,a],[on,e,d],[on,d,f],[on,a,e],[on,b,table],[on,c,table],[on,f,table]], [[clear,c],[clear,b],[on,e,d],[on,d,f],[on,a,e],[on,b,a],[on,c,table],[on,f,table]], [[clear,a],[clear,b],[on,e,d],[on,d,f],[on,a,e],[on,b,c],[on,c,table],[on,f,table]], [[clear,a],[clear,e],[on,e,d],[on,d,f],[on,a,b],[on,b,c],[on,c,table],[on,f,table]], [[clear,d],[clear,e],[on,e,a],[on,d,f],[on,a,b],[on,b,c],[on,c,table],[on,f,table]] Steps: 1640
Notes	The algorithm successfully found a solution, but the path length was very long (1640 steps), as uninformed DFS is not efficient for this task.
Updated Notes	IDDFS Path: (Length: 5) [on,e,d],[on,d,a],[on,a,table],[on,b,c],[on,c,table],[on,f,table],[clear,e],[clear,b],[clear,f]], [clear,d],[on,e,table],[on,d,a],[on,a,table],[on,b,c],[on,c,table],[on,f,table],[clear,e],[clear,b],[clear,f]], [clear,d],[on,e,table],[on,d,f],[on,a,table],[on,b,c],[on,c,table],[on,f,table],[clear,e],[clear,b],[clear,a]], [clear,d],[on,e,table],[on,d,f],[on,a,b],[on,b,c],[on,c,table],[on,f,table],[clear,e],[clear,a]], [clear,d],[on,e,a],[on,d,f],[on,a,b],[on,b,c],[on,c,table],[on,f,table],[clear,e]] <div> <div>state[0]</div> <div>state[1]</div> <div>state[2]</div> <div>state[3]</div> <div>state[4]</div> </div> <div> <div>E</div> <div>D B</div> <div>A C F</div> <div>E A C F</div> <div>E A C F</div> <div>E C F</div> <div>E</div> <div>A</div> <div>B D</div> <div>C F</div> </div>

TEST 4: (5 blocks)	
Start	start([on,e,b],[on,b,a],[on,a,table],[on,c,table],[on,d,table],[clear,e],[clear,c],[clear,d])). <div> <div>E</div> <div>B</div> <div>A C D</div> </div>
Goal	goal([on,e,a],[on,a,c],[on,c,d],[on,d,table],[on,b,table],[clear,b],[clear,e])). <div> <div>E</div> <div>A</div> <div>C</div> <div>B D</div> </div>
Notes	The program got stuck and couldn't find a solution. Since the DFS wasn't effectively finding

	solutions because it was searching too deep, I changed it to an Iterative deepening DFS to improve it, and find shorter paths.
<u>Path (IDDFS)</u>	<pre> [[on,e,b],[on,b,a],[on,a,table],[on,c,table],[on,d,table],[clear,e],[clear,c],[clear,d]], [[clear,b],[on,e,table],[on,b,a],[on,a,table],[on,c,table],[on,d,table],[clear,e],[clear,c],[clear,d]], [[clear,a],[clear,b],[on,e,table],[on,b,table],[on,a,table],[on,c,table],[on,d,table],[clear,e],[clear,c],[clear,d]], [[clear,a],[clear,b],[on,e,table],[on,b,table],[on,a,table],[on,c,d],[on,d,table],[clear,e],[clear,c]], [[clear,a],[clear,b],[on,e,table],[on,b,table],[on,a,c],[on,c,d],[on,d,table],[clear,e]], [[clear,b],[on,e,a],[on,b,table],[on,a,c],[on,c,d],[on,d,table],[clear,e]], </pre> <p>Path Length: 6</p>
	<div> <div>state[0]</div> <div> <div>E</div> <div>B</div> <div>A C D</div> </div> </div> <div> <div>state[1]</div> <div> <div>B</div> <div>E A C D</div> </div> </div> <div> <div>state[2]</div> <div> <div>E B A C D</div> </div> </div> <div> <div>state[3]</div> <div> <div>C</div> <div>E B A D</div> </div> </div> <div> <div>state[4]</div> <div> <div>A</div> <div>C</div> <div>E B D</div> </div> </div> <div> <div>state[5]</div> <div> <div>E</div> <div>A</div> <div>C</div> <div>B D</div> </div> </div>
<u>Updated Notes</u>	With IDDFS, the program is more efficient in finding a solution and the shortest path is returned.

GitHub Code:

https://github.com/elys33ve/AI-principles-CS4033/blob/main/BlocksWorld/assignment3_code.pl