Assignment 4: Blocks World

Huy Nguyen, Toan Ly, Toan Nham (All contributed equally)

```
run_all.
Running all test cases...
--------------------------------------------------------------------------------
Running Test Case 1
--------------------------------------------------------------------------------
Start State: [[on,a,b],[on,b,table],[clear,a]]
Goal State: [[on,b,a],[on,a,table],[clear,b]]
Processing...
Finish!
Path:
[[on,a,table],[on,b,table],[clear,a],[clear,b]]
[[on,a,table],[on,b,a],[clear,b]]
Final State: [[on,a,table],[on,b,a],[clear,b]]
--------------------------------------------------------------------------------
=> Execution Time: 0 ms
--------------------------------------------------------------------------------
```

```
?- run_all.
```

Comment: The states to the final solution are correct and optimal using minimal steps.

```
--------------------------------------------------------------------------------
Running Test Case 2
--------------------------------------------------------------------------------
Start State: [[on,a,table],[on,b,table],[clear,a],[clear,b]]
Goal State: [[on,a,b],[on,b,table],[clear,a]]
Processing...
Finish!
Path:
[[on,a,b],[on,b,table],[clear,a]]
Final State: [[on,a,b],[on,b,table],[clear,a]]
--------------------------------------------------------------------------------
=> Execution Time: 0 ms
--------------------------------------------------------------------------------
```

```
?- run_all.
```

Comment: The path is straightforward and correct to reach to the final state

```
--------------------------------------------------------------------------
Running Test Case 3
--------------------------------------------------------------------------
Start State: [[on,a,b],[on,b,table],[on,c,table],[clear,a],[clear,c]]
Goal State: [[on,c,a],[on,a,b],[on,b,table],[clear,c]]
Processing...
Finish!
Path:
[[on,a,c],[on,b,table],[on,c,table],[clear,a],[clear,b]]
[[on,a,b],[on,b,table],[on,c,table],[clear,a],[clear,c]]
[[on,a,table],[on,b,table],[on,c,table],[clear,a],[clear,c],[clear,b]]
[[on,a,table],[on,b,table],[on,c,a],[clear,c],[clear,b]]
[[on,a,table],[on,b,table],[on,c,b],[clear,c],[clear,a]]
[[on,a,table],[on,b,table],[on,c,table],[clear,c],[clear,a],[clear,b]]
[[on,a,b],[on,b,table],[on,c,table],[clear,c],[clear,a]]
[[on,a,b],[on,b,table],[on,c,a],[clear,c]]
Final State: [[on,a,b],[on,b,table],[on,c,a],[clear,c]]
--------------------------------------------------------------------------
=> Execution Time: 6 ms
--------------------------------------------------------------------------
```

```
?- run_all.
```

Comment: Although the solution is correct, the paths are not optimal as there should be only 1 step to reach the final solution. It may be because dfs starts with block a first and moves it around instead of just putting block c on top of a.

```
------------------------------------------------------------------------
Running Test Case 4
------------------------------------------------------------------------
Start State: [[on,a,b],[on,b,table],[on,c,table],[clear,a],[clear,c]]
Goal State: [[on,c,b],[on,a,c],[on,b,table],[clear,a]]
Processing...
Finish!
Path:
[[on,a,c],[on,b,table],[on,c,table],[clear,a],[clear,b]]
[[on,a,b],[on,b,table],[on,c,table],[clear,a],[clear,c]]
[[on,a,table],[on,b,table],[on,c,table],[clear,a],[clear,c],[clear,b]]
[[on,a,table],[on,b,table],[on,c,a],[clear,c],[clear,b]]
[[on,a,table],[on,b,table],[on,c,b],[clear,c],[clear,a]]
[[on,a,table],[on,b,table],[on,c,table],[clear,c],[clear,a],[clear,b]]
[[on,a,b],[on,b,table],[on,c,table],[clear,c],[clear,a]]
[[on,a,c],[on,b,table],[on,c,table],[clear,b],[clear,a]]
[[on,a,table],[on,b,table],[on,c,table],[clear,b],[clear,a],[clear,c]]
[[on,a,table],[on,b,a],[on,c,table],[clear,b],[clear,c]]
[[on,a,table],[on,b,c],[on,c,table],[clear,b],[clear,a]]
[[on,a,b],[on,b,c],[on,c,table],[clear,a]]
[[on,a,table],[on,b,c],[on,c,table],[clear,a],[clear,b]]
[[on,a,table],[on,b,table],[on,c,table],[clear,a],[clear,b],[clear,c]]
[[on,a,table],[on,b,table],[on,c,a],[clear,b],[clear,c]]
[[on,a,table],[on,b,table],[on,c,b],[clear,a],[clear,c]]
[[on,a,c],[on,b,table],[on,c,b],[clear,a]]
Final State: [[on,a,c],[on,b,table],[on,c,b],[clear,a]]
------------------------------------------------------------------------
=> Execution Time: 1 ms
------------------------------------------------------------------------
```

```
?- run_all.
```

Comment: Similarly, although the final state is correct, the paths are not optimal since the algorithm starts putting block a on top of other blocks instead of on the table, which will reach the goal faster.

```
--------------------------------------------------------------------------
Running Test Case 5
--------------------------------------------------------------------------
Start State: [[on,a,b],[on,b,c],[on,c,table],[clear,a]]
Goal State: [[on,b,table],[on,c,a],[on,a,table],[clear,c],[clear,b]]
Processing...
Finish!
Path:
[[on,a,table],[on,b,c],[on,c,table],[clear,a],[clear,b]]
[[on,a,table],[on,b,a],[on,c,table],[clear,c],[clear,b]]
[[on,a,table],[on,b,table],[on,c,table],[clear,c],[clear,b],[clear,a]]
[[on,a,table],[on,b,table],[on,c,b],[clear,c],[clear,a]]
[[on,a,table],[on,b,table],[on,c,a],[clear,c],[clear,b]]
Final State: [[on,a,table],[on,b,table],[on,c,a],[clear,c],[clear,b]]
--------------------------------------------------------------------------
=> Execution Time: 1 ms
--------------------------------------------------------------------------
```

```
?- run_all.
```

Comment: Similarly, the solution is correct but not optimal. It seems that the algorithm will only put a block onto the table if there're no more blocks to be put onto.

```
--------------------------------------------------------------------------
Running Test Case 6
--------------------------------------------------------------------------
Start State: [[on,a,b],[on,b,c],[on,c,table],[clear,a],[on,d,table],[clear,d]]
Goal State: [[on,c,b],[on,d,c],[on,a,d],[on,b,table],[clear,a]]
Processing...
Finish!
Path:
[[on,a,d],[on,b,c],[on,c,table],[clear,a],[on,d,table],[clear,b]]
[[on,a,b],[on,b,c],[on,c,table],[clear,a],[on,d,table],[clear,d]]
[[on,a,table],[on,b,c],[on,c,table],[clear,a],[on,d,table],[clear,d],[clear,b]]
[[on,a,table],[on,b,a],[on,c,table],[clear,c],[on,d,table],[clear,d],[clear,b]]
[[on,a,table],[on,b,d],[on,c,table],[clear,c],[on,d,table],[clear,a],[clear,b]]
[[on,a,table],[on,b,c],[on,c,table],[clear,d],[on,d,table],[clear,a],[clear,b]]
[[on,a,table],[on,b,a],[on,c,table],[clear,d],[on,d,table],[clear,c],[clear,b]]
[[on,a,table],[on,b,d],[on,c,table],[clear,a],[on,d,table],[clear,c],[clear,b]]
[[on,a,table],[on,b,table],[on,c,table],[clear,a],[on,d,table],[clear,c],[clear,b],[clear,d]]
[[on,a,c],[on,b,table],[on,c,table],[clear,a],[on,d,table],[clear,b],[clear,d]]
[[on,a,b],[on,b,table],[on,c,table],[clear,a],[on,d,table],[clear,c],[clear,d]]
[[on,a,d],[on,b,table],[on,c,table],[clear,a],[on,d,table],[clear,c],[clear,b]]
[[on,a,c],[on,b,table],[on,c,table],[clear,a],[on,d,table],[clear,d],[clear,b]]
[[on,a,b],[on,b,table],[on,c,table],[clear,a],[on,d,table],[clear,d],[clear,c]]
[[on,a,d],[on,b,table],[on,c,table],[clear,a],[on,d,table],[clear,b],[clear,c]]
[[on,a,table],[on,b,table],[on,c,table],[clear,a],[on,d,table],[clear,b],[clear,c],[clear,d]]
[[on,a,table],[on,b,a],[on,c,table],[on,d,table],[clear,b],[clear,c],[clear,d]]
[[on,a,table],[on,b,c],[on,c,table],[on,d,table],[clear,b],[clear,a],[clear,d]]
[[on,a,table],[on,b,d],[on,c,table],[on,d,table],[clear,b],[clear,a],[clear,c]]
[[on,a,table],[on,b,a],[on,c,table],[on,d,table],[clear,b],[clear,d],[clear,c]]
[[on,a,table],[on,b,c],[on,c,table],[on,d,table],[clear,b],[clear,d],[clear,a]]
[[on,a,table],[on,b,d],[on,c,table],[on,d,table],[clear,b],[clear,c],[clear,a]]
[[on,a,table],[on,b,table],[on,c,table],[on,d,table],[clear,b],[clear,c],[clear,a],[clear,d]]
[[on,a,table],[on,b,table],[on,c,b],[on,d,table],[clear,c],[clear,a],[clear,d]]
[[on,a,table],[on,b,table],[on,c,a],[on,d,table],[clear,c],[clear,b],[clear,d]]
[[on,a,table],[on,b,table],[on,c,d],[on,d,table],[clear,c],[clear,b],[clear,a]]
[[on,a,table],[on,b,table],[on,c,b],[on,d,table],[clear,c],[clear,d],[clear,a]]
[[on,a,table],[on,b,table],[on,c,a],[on,d,table],[clear,c],[clear,d],[clear,b]]
[[on,a,table],[on,b,table],[on,c,d],[on,d,table],[clear,c],[clear,a],[clear,b]]
[[on,a,table],[on,b,table],[on,c,table],[on,d,table],[clear,c],[clear,a],[clear,b],[clear,d]]
[[on,a,c],[on,b,table],[on,c,table],[on,d,table],[clear,a],[clear,b],[clear,d]]
[[on,a,b],[on,b,table],[on,c,table],[on,d,table],[clear,a],[clear,c],[clear,d]]
[[on,a,d],[on,b,table],[on,c,table],[on,d,table],[clear,a],[clear,c],[clear,b]]
[[on,a,c],[on,b,table],[on,c,table],[on,d,table],[clear,a],[clear,d],[clear,b]]
[[on,a,b],[on,b,table],[on,c,table],[on,d,table],[clear,a],[clear,d],[clear,c]]
[[on,a,d],[on,b,table],[on,c,table],[on,d,table],[clear,a],[clear,b],[clear,c]]
[[on,a,table],[on,b,table],[on,c,table],[on,d,table],[clear,a],[clear,b],[clear,c],[clear,d]]
[[on,a,table],[on,b,c],[on,c,table],[on,d,table],[clear,a],[clear,b],[clear,d]]
[[on,a,table],[on,b,a],[on,c,table],[on,d,table],[clear,c],[clear,b],[clear,d]]
[[on,a,table],[on,b,d],[on,c,table],[on,d,table],[clear,c],[clear,b],[clear,a]]
[[on,a,table],[on,b,c],[on,c,table],[on,d,table],[clear,d],[clear,b],[clear,a]]
[[on,a,table],[on,b,a],[on,c,table],[on,d,table],[clear,d],[clear,b],[clear,c]]
[[on,a,table],[on,b,d],[on,c,table],[on,d,table],[clear,a],[clear,b],[clear,c]]
[[on,a,b],[on,b,d],[on,c,table],[on,d,table],[clear,a],[clear,c]]
[[on,a,c],[on,b,d],[on,c,table],[on,d,table],[clear,a],[clear,b]]
[[on,a,c],[on,b,a],[on,c,table],[on,d,table],[clear,d],[clear,b]]
[[on,a,c],[on,b,table],[on,c,table],[on,d,table],[clear,d],[clear,b],[clear,a]]
[[on,a,d],[on,b,table],[on,c,table],[on,d,table],[clear,c],[clear,b],[clear,a]]
[[on,a,b],[on,b,table],[on,c,table],[on,d,table],[clear,c],[clear,d],[clear,a]]
[[on,a,c],[on,b,table],[on,c,table],[on,d,table],[clear,b],[clear,d],[clear,a]]
[[on,a,d],[on,b,table],[on,c,table],[on,d,table],[clear,b],[clear,c],[clear,a]]
[[on,a,b],[on,b,table],[on,c,table],[on,d,table],[clear,d],[clear,c],[clear,a]]
[[on,a,table],[on,b,table],[on,c,table],[on,d,table],[clear,d],[clear,c],[clear,a],[clear,b]]
[[on,a,table],[on,b,table],[on,c,table],[on,d,c],[clear,d],[clear,a],[clear,b]]
[[on,a,table],[on,b,table],[on,c,table],[on,d,a],[clear,d],[clear,c],[clear,b]]
```

```
[[on,a,table],[on,b,table],[on,c,table],[on,d,a],[clear,d],[clear,c],[clear,b]]
[[on,a,table],[on,b,table],[on,c,table],[on,d,b],[clear,d],[clear,c],[clear,a]]
[[on,a,table],[on,b,table],[on,c,table],[on,d,c],[clear,d],[clear,b],[clear,a]]
[[on,a,table],[on,b,table],[on,c,table],[on,d,a],[clear,d],[clear,b],[clear,c]]
[[on,a,table],[on,b,table],[on,c,table],[on,d,b],[clear,d],[clear,a],[clear,c]]
[[on,a,table],[on,b,table],[on,c,table],[on,d,table],[clear,d],[clear,a],[clear,c],[clear,b]]
[[on,a,c],[on,b,table],[on,c,table],[on,d,table],[clear,d],[clear,a],[clear,b]]
[[on,a,d],[on,b,table],[on,c,table],[on,d,table],[clear,c],[clear,a],[clear,b]]
[[on,a,b],[on,b,table],[on,c,table],[on,d,table],[clear,c],[clear,a],[clear,d]]
[[on,a,c],[on,b,table],[on,c,table],[on,d,table],[clear,b],[clear,a],[clear,d]]
[[on,a,d],[on,b,table],[on,c,table],[on,d,table],[clear,b],[clear,a],[clear,c]]
[[on,a,b],[on,b,table],[on,c,table],[on,d,table],[clear,d],[clear,a],[clear,c]]
[[on,a,b],[on,b,table],[on,c,table],[on,d,a],[clear,d],[clear,c]]
[[on,a,b],[on,b,table],[on,c,table],[on,d,c],[clear,d],[clear,a]]
[[on,a,d],[on,b,table],[on,c,table],[on,d,c],[clear,b],[clear,a]]
[[on,a,table],[on,b,table],[on,c,table],[on,d,c],[clear,b],[clear,a],[clear,d]]
[[on,a,table],[on,b,table],[on,c,table],[on,d,b],[clear,c],[clear,a],[clear,d]]
[[on,a,table],[on,b,table],[on,c,table],[on,d,a],[clear,c],[clear,b],[clear,d]]
[[on,a,table],[on,b,table],[on,c,table],[on,d,c],[clear,a],[clear,b],[clear,d]]
[[on,a,table],[on,b,table],[on,c,table],[on,d,b],[clear,a],[clear,c],[clear,d]]
[[on,a,table],[on,b,table],[on,c,table],[on,d,a],[clear,b],[clear,c],[clear,d]]
[[on,a,table],[on,b,table],[on,c,table],[on,d,table],[clear,b],[clear,c],[clear,d],[clear,a]]
[[on,a,table],[on,b,table],[on,c,d],[on,d,table],[clear,b],[clear,c],[clear,a]]
[[on,a,table],[on,b,table],[on,c,b],[on,d,table],[clear,d],[clear,c],[clear,a]]
[[on,a,table],[on,b,table],[on,c,a],[on,d,table],[clear,d],[clear,c],[clear,b]]
[[on,a,table],[on,b,table],[on,c,d],[on,d,table],[clear,a],[clear,c],[clear,b]]
[[on,a,table],[on,b,table],[on,c,b],[on,d,table],[clear,a],[clear,c],[clear,d]]
[[on,a,table],[on,b,table],[on,c,a],[on,d,table],[clear,b],[clear,c],[clear,d]]
[[on,a,table],[on,b,c],[on,c,a],[on,d,table],[clear,b],[clear,d]]
[[on,a,table],[on,b,d],[on,c,a],[on,d,table],[clear,b],[clear,c]]
[[on,a,table],[on,b,d],[on,c,b],[on,d,table],[clear,a],[clear,c]]
[[on,a,table],[on,b,d],[on,c,table],[on,d,table],[clear,a],[clear,c],[clear,b]]
[[on,a,table],[on,b,a],[on,c,table],[on,d,table],[clear,d],[clear,c],[clear,b]]
[[on,a,table],[on,b,c],[on,c,table],[on,d,table],[clear,d],[clear,a],[clear,b]]
[[on,a,table],[on,b,d],[on,c,table],[on,d,table],[clear,c],[clear,a],[clear,b]]
[[on,a,table],[on,b,a],[on,c,table],[on,d,table],[clear,c],[clear,d],[clear,b]]
[[on,a,table],[on,b,c],[on,c,table],[on,d,table],[clear,a],[clear,d],[clear,b]]
[[on,a,table],[on,b,table],[on,c,table],[on,d,table],[clear,a],[clear,d],[clear,b],[clear,c]]
[[on,a,table],[on,b,table],[on,c,table],[on,d,b],[clear,a],[clear,d],[clear,c]]
[[on,a,table],[on,b,table],[on,c,table],[on,d,a],[clear,b],[clear,d],[clear,c]]
[[on,a,table],[on,b,table],[on,c,table],[on,d,c],[clear,b],[clear,d],[clear,a]]
[[on,a,table],[on,b,table],[on,c,table],[on,d,a],[clear,c],[clear,d],[clear,b]]
[[on,a,table],[on,b,table],[on,c,d],[on,d,a],[clear,c],[clear,b]]
[[on,a,table],[on,b,table],[on,c,b],[on,d,a],[clear,c],[clear,d]]
[[on,a,table],[on,b,table],[on,c,b],[on,d,c],[clear,a],[clear,d]]
[[on,a,table],[on,b,table],[on,c,b],[on,d,table],[clear,a],[clear,d],[clear,c]]
[[on,a,table],[on,b,table],[on,c,a],[on,d,table],[clear,b],[clear,d],[clear,c]]
[[on,a,table],[on,b,table],[on,c,d],[on,d,table],[clear,b],[clear,a],[clear,c]]
[[on,a,table],[on,b,table],[on,c,b],[on,d,table],[clear,d],[clear,a],[clear,c]]
[[on,a,table],[on,b,table],[on,c,b],[on,d,a],[clear,d],[clear,c]]
[[on,a,table],[on,b,table],[on,c,b],[on,d,c],[clear,d],[clear,a]]
[[on,a,d],[on,b,table],[on,c,b],[on,d,c],[clear,a]]
Final State: [[on,a,d],[on,b,table],[on,c,b],[on,d,c],[clear,a]]
---------------------------------------------------------------------------
=> Execution Time: 8 ms
---------------------------------------------------------------------------

All test cases passed!
```

Comment: This is our last test case, the final state is correct, but the paths are not optimal because of similar reason explained above.