Take-Home Practice on Search - Solutions

The 8-puzzle problem is a puzzle invented and popularized by Noyes Palmer Chapman in the 1870s. It is played on a 3-by-3 grid with 8 square blocks labeled 1 through 8 and a blank square. Your goal is to rearrange the blocks so that they are in order. You are permitted to slide blocks horizontally or vertically into the blank square. The following shows a sequence of legal moves from an initial board position (left) to the goal position (right).

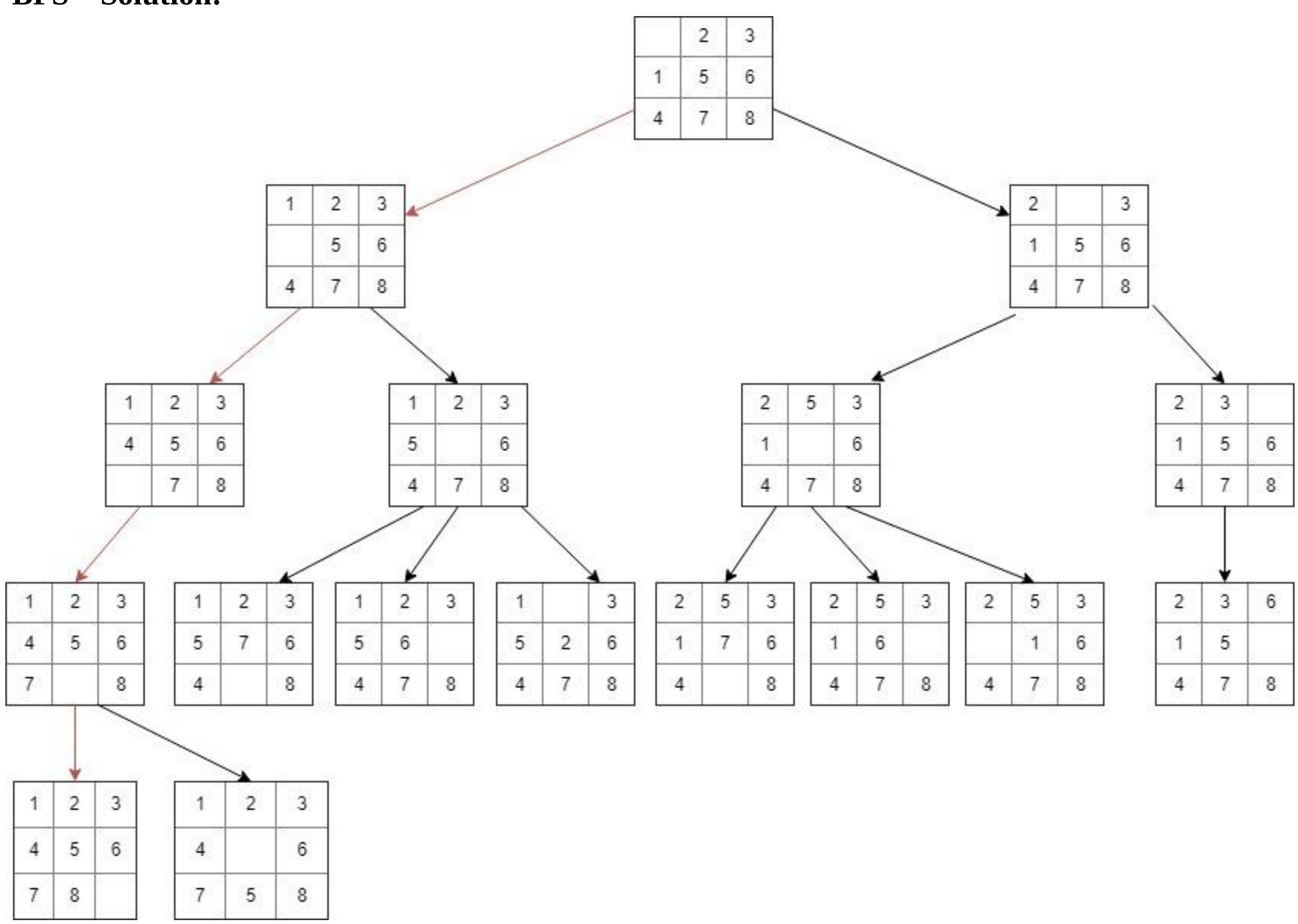
```
   1  3        1     3        1  2  3        1  2  3        1  2  3
4  2  5   =>   4  2  5   =>   4     5   =>   4  5      =>   4  5  6
7  8  6        7  8  6        7  8  6        7  8  6        7  8

initial                                                     goal
```

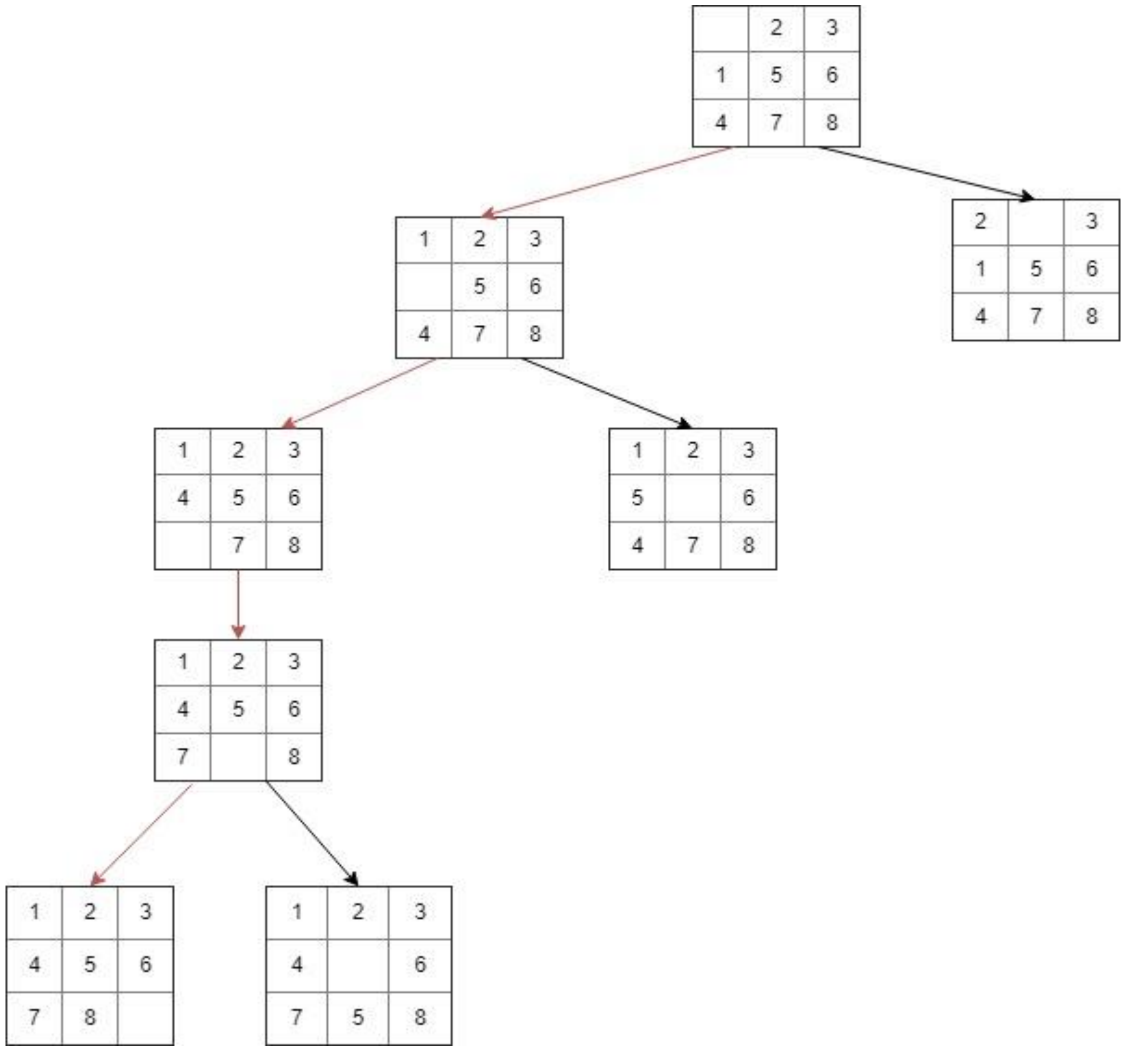
1. Start with the initial configuration below, construct a search space tree using BFS and DFS to reach the goal state as defined above. Proceed in the order down-right-up-left for expansion.

_ 2 3
1 5 6
4 7 8

BFS – Solution:



DFS – Solution:



Red lines indicate traversal, black lines indicate expansion

2. Apply the Greedy Best First Search algorithm to the configuration given below using the heuristic function f(n) = number of misplaced tiles. Determine the number of moves needed to reach the goal state mentioned below using this heuristic.

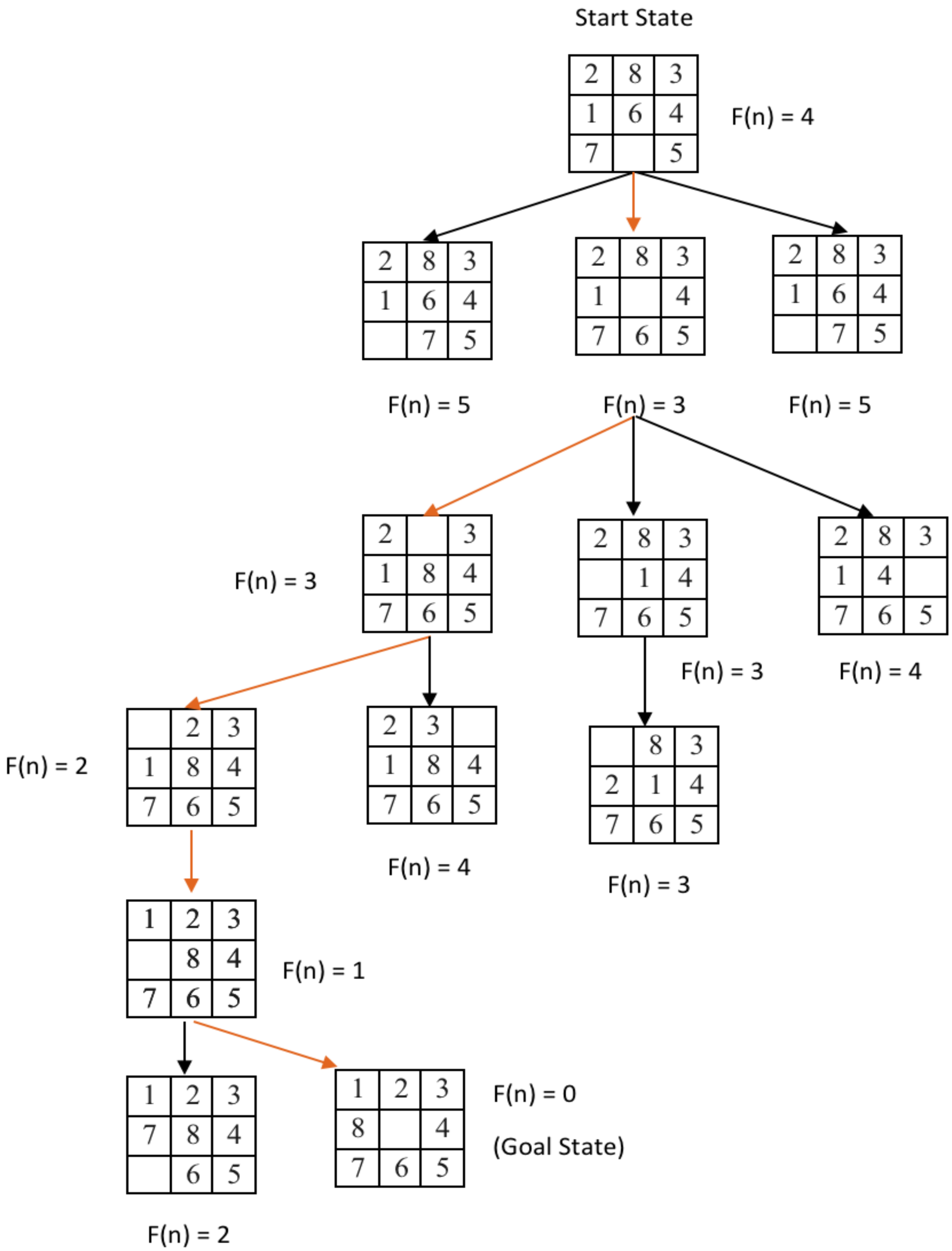
Start State

2	8	3
1	6	4
7		5

Goal State

1	2	3
8		4
7	6	5

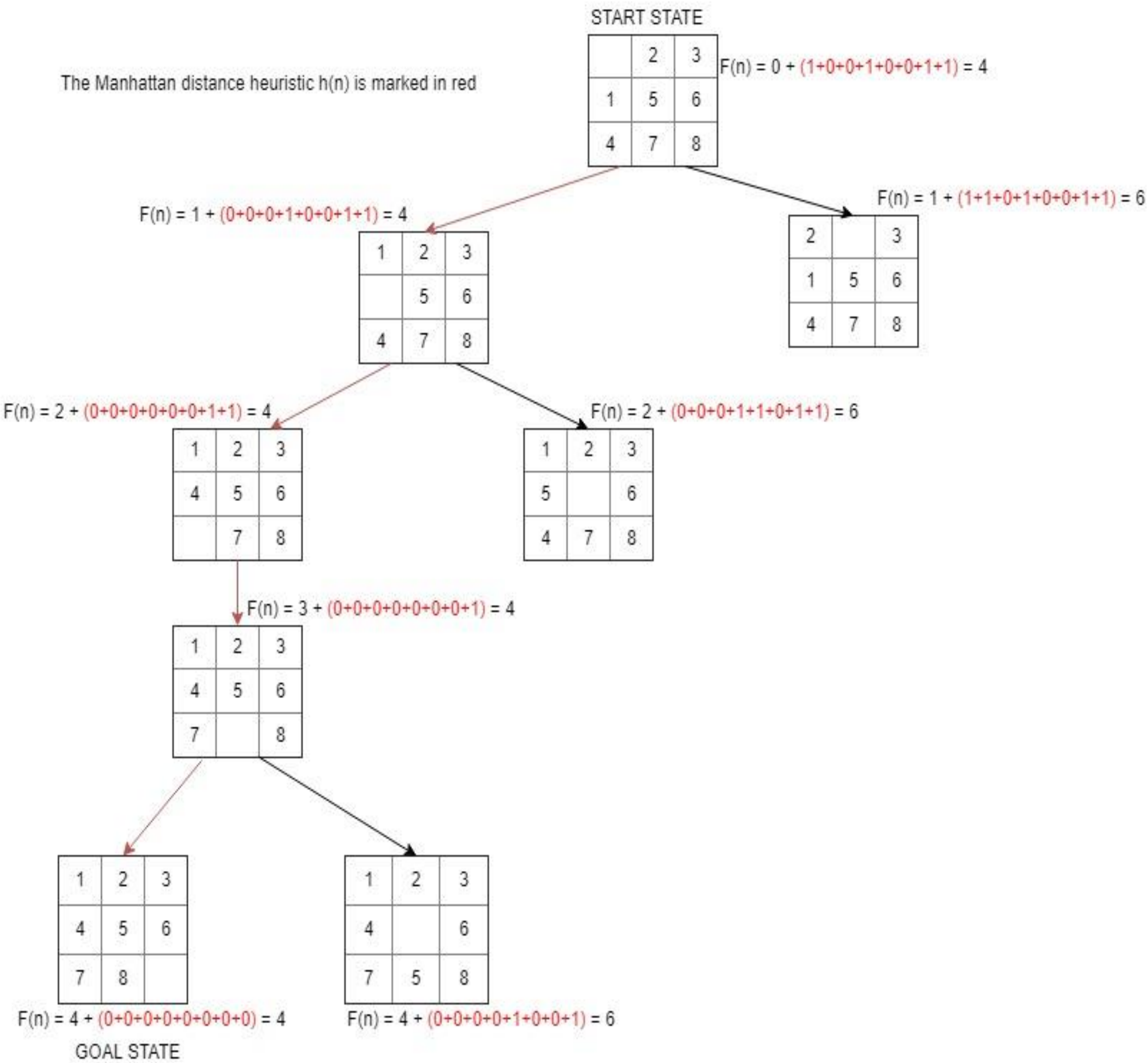
Solution: 5 moves. The state tree and moves are shown below.


Start State
F(n) = 4
F(n) = 5
F(n) = 3
F(n) = 5
F(n) = 3
F(n) = 3
F(n) = 4
F(n) = 2
F(n) = 4
F(n) = 3
F(n) = 1
F(n) = 0
(Goal State)
F(n) = 2

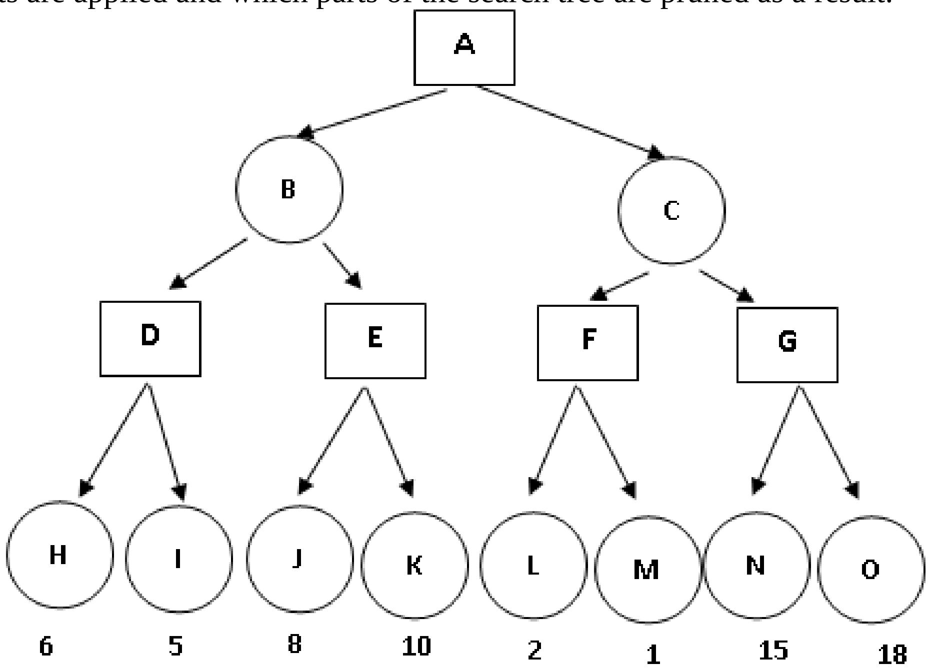
3. Using Manhattan distance as a heuristic, perform A* search to reach goal state. Use the same initial configuration as (1).

Solution:

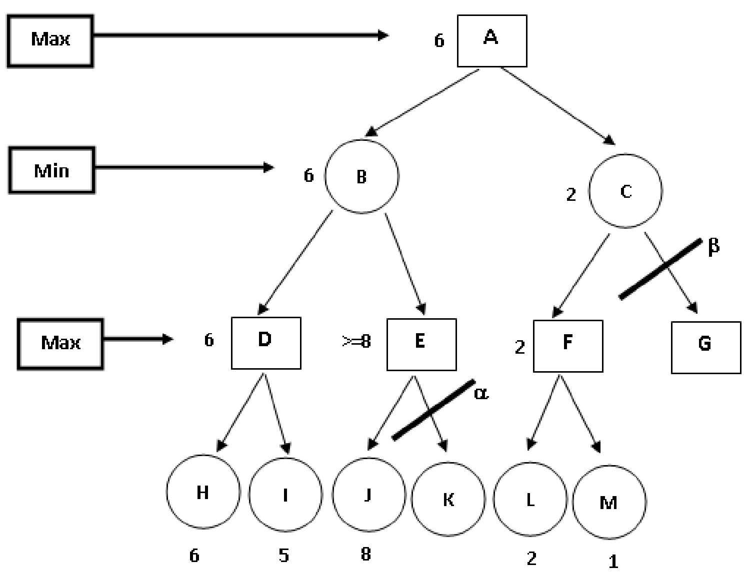
F(N) = the distance of the node from the source + the Manhattan distance of each tile from its target location



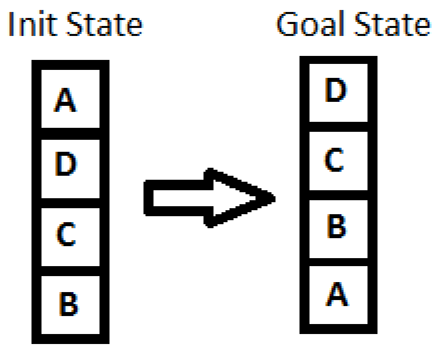
4. Consider the search tree below:
   a. Write out the values of each node using minmax algorithm (circles are MIN nodes) on the following search tree,
   b. Apply the alpha-beta pruning algorithm to the same tree and show the search tree that would be built by this algorithm. Make sure that you show where the alpha and beta cuts are applied and which parts of the search tree are pruned as a result.



Solution:



5. Consider a block-world problem where similar and equal blocks (A to D) are given as shown here. They are arranged in the initial state and need to be arranged as in the goal state.



The following two operators are allowed:

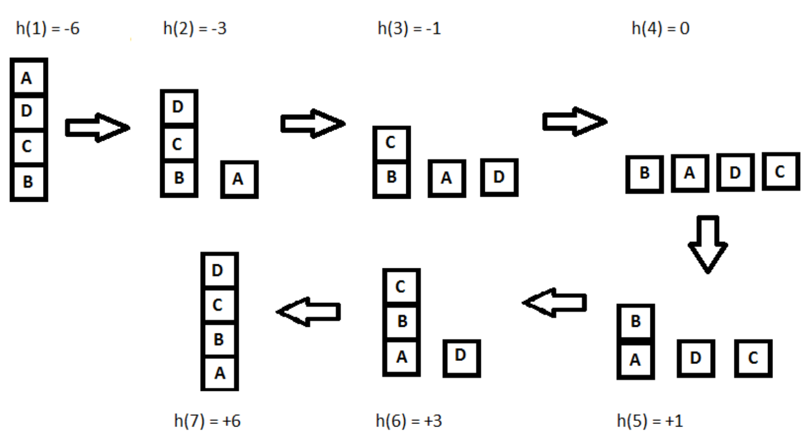
R1: Pick up one block and put it on the table
R2: Pick up one block and put it on another.

There is no restriction on how many blocks can be on the table at any given time. However you can only use R1 or R2 at each state.

Let the heuristic function be defined in the following way**:**
For each block in the structure +1 for each block underneath it that is correctly positioned and -1 for each block underneath it that is incorrectly positioned. If there are no blocks underneath, add 0.

So for the initial state, the heuristic value is -6 because A has three incorrectly placed blocks underneath D has two and C has one. -3 -2 -1 = -6

Apply hill-climbing using the above heuristic and determine the number of moves needed to reach the goal state.



Solution: