Prolog Programming Assignment #2: State Space Problem Solving

Learning Abstract

Prolog Programming Assignment #2 is made up of 9 tasks. The 9 tasks of the assignment involve the Towers of Hanoi problem and trying to solve it Prolog , using unit testing and predicates. Task 1-2 involve setting up the program I prolog and understanding the material before beginning on task 3. Task 3-8 help you get to the complete solution of the problem. In task 9 we are asked to show all of our source code.

Task 1: One Move Predicate and a Unit Test

State Space Operator Code:

```
55  m12([Tower1Before,Tower2Before,Tower3],[Tower1After,Tower2After,Tower3]) :-
56      Tower1Before = [H|T],
57      Tower1After = T,
58      Tower2Before = L,
59      Tower2After = [H|L].
```

Unit Test Code:

```
61  test__m12 :-
62      write('Testing: move_m12\n'),
63      TowersBefore = [[t,s,m,l,h],[],[]],
64      trace('','TowersBefore',TowersBefore),
65      m12(TowersBefore,TowersAfter),
66      trace('','TowersAfter',TowersAfter).
```

Unit Test Demo:

```
1 ?- consult('toh.pro').
true.

2 ?-  test__m12.
Testing: move_m12
TowersBefore = [[t,s,m,l,h],[],[]]
TowersAfter = [[s,m,l,h],[t],[]]
true.
```

Task 4: The Remaining Five Move Predicates and a Unit Tests

State Space Operator Code:

```
55  m12([Tower1Before,Tower2Before,Tower3],[Tower1After,Tower2After,Tower3]) :-
56      Tower1Before = [H|T],
57      Tower1After = T,
58      Tower2Before = L,
59      Tower2After = [H|L].
60
61  m13([Tower1Before,Tower2,Tower3Before],[Tower1After,Tower2,Tower3After]) :-
62      Tower1Before = [H|T],
63      Tower1After = T,
64      Tower3Before = L,
65      Tower3After = [H|L].
66
67  m21([Tower1Before,Tower2Before,Tower3],[Tower1After,Tower2After,Tower3]) :-
68      Tower2Before = [H|T],
69      Tower2After = T,
70      Tower1Before = L,
71      Tower1After = [H|L].
72
73  m23([Tower1,Tower2Before,Tower3Before],[Tower1,Tower2After,Tower3After]) :-
74      Tower2Before = [H|T],
75      Tower2After = T,
76      Tower3Before = L,
77      Tower3After = [H|L].
78
79  m31([Tower1Before,Tower2,Tower3Before],[Tower1After,Tower2,Tower3After]) :-
80      Tower3Before = [H|T],
81      Tower3After = T,
82      Tower1Before = L,
83      Tower1After = [H|L].
84
85  m32([Tower1,Tower2Before,Tower3Before],[Tower1,Tower2After,Tower3After]) :-
86      Tower3Before = [H|T],
87      Tower3After = T,
88      Tower2Before = L,
89      Tower2After = [H|L].
```

Unit Test Code:

```
91  test__m12 :-
92      write('Testing: move_m12\n'),
93      TowersBefore = [[t,s,m,l,h],[],[]],
94      trace('','TowersBefore',TowersBefore),
95      m12(TowersBefore,TowersAfter),
96      trace('','TowersAfter',TowersAfter).
97
98  test__m13 :-
99      write('Testing: move_m13\n'),
100     TowersBefore = [[t,s,m,l,h],[],[]],
101     trace('','TowersBefore',TowersBefore),
102     m13(TowersBefore,TowersAfter),
103     trace('','TowersAfter',TowersAfter).
104
105 test__m21 :-
106     write('Testing: move_m21\n'),
107     TowersBefore = [[m,l,h],[t,s],[]],
108     trace('','TowersBefore',TowersBefore),
109     m21(TowersBefore,TowersAfter),
110     trace('','TowersAfter',TowersAfter).
111
112 test__m23 :-
113     write('Testing: move_m23\n'),
114     TowersBefore = [[l,h],[t,s],[m]],
115     trace('','TowersBefore',TowersBefore),
116     m23(TowersBefore,TowersAfter),
117     trace('','TowersAfter',TowersAfter).
118
119 test__m31 :-
120     write('Testing: move_m31\n'),
121     TowersBefore = [[m,l,h],[],[t,s]],
122     trace('','TowersBefore',TowersBefore),
123     m31(TowersBefore,TowersAfter),
124     trace('','TowersAfter',TowersAfter).
125
126 test__m32 :-
127     write('Testing: move_m32\n'),
128     TowersBefore = [[l,h],[m],[t,s]],
129     trace('','TowersBefore',TowersBefore),
130     m32(TowersBefore,TowersAfter),
131     trace('','TowersAfter',TowersAfter).
```

Unit Test Demo:

```
Testing: move_m12
TowersBefore = [[t,s,m,l,h],[],[]]
TowersAfter = [[s,m,l,h],[t],[]]
true.

3 ?- test__m13.
Testing: move_m13
TowersBefore = [[t,s,m,l,h],[],[]]
TowersAfter = [[s,m,l,h],[],[t]]
true.

4 ?- test__m21.
Testing: move_m21
TowersBefore = [[m,l,h],[t,s],[]]
TowersAfter = [[t,m,l,h],[s],[]]
true.

5 ?- test__m23.
Testing: move_m23
TowersBefore = [[l,h],[t,s],[m]]
TowersAfter = [[l,h],[s],[t,m]]
true.

6 ?- test__m31.
Testing: move_m31
TowersBefore = [[m,l,h],[],[t,s]]
TowersAfter = [[t,m,l,h],[],[s]]
true.

7 ?- test__m32.
Testing: move_m32
TowersBefore = [[l,h],[m],[t,s]]
TowersAfter = [[l,h],[t,m],[s]]
true.
```

Task 5: Valid State Predicate and Unit Test

Predicate:

```
147  valid_state([A|[B|[C]]]) :- state(A), state(B), state(C).
148      state([]).
149      state([t]).
150      state([s]).
151      state([m]).
152      state([l]).
153      state([h]).
154      state([t,s]).
155      state([t,s,m]).
156      state([t,s,m,l]).
157      state([t,s,l]).
158      state([t,s,l,h]).
159      state([t,s,h]).
160      state([t,s,m,h]).
161      state([t,m]).
162      state([t,m,l]).
163      state([t,m,l,h]).
164      state([t,m,h]).
165      state([t,l,h]).
166      state([t,l]).
167      state([t,h]).
168      state([t,s,m,l,h]).
169      state([s]).
170      state([s,m]).
171      state([s,m,l]).
172      state([s,m,l,h]).
173      state([s,l]).
174      state([s,l,h]).
175      state([s,h]).
176      state([s,m,h]).
177      state([m]).
178      state([m,l]).
179      state([m,l,h]).
180      state([m,h]).
181      state([l]).
182      state([l,h]).
183      state([h]).
```

Unit Test Code:

```
133  test__valid_state :-
134      write('Testing: valid_state\n'),
135      test__vs([[l,t,s,m,h],[],[]]),
136      test__vs([[t,s,m,l,h],[],[]]),
137      test__vs([[],[h,t,s,m],[l]]),
138      test__vs([[],[t,s,m,h],[l]]),
139      test__vs([[],[h],[l,m,s,t]]),
140      test__vs([[],[h],[t,s,m,l]]).
141      test__vs(S) :-
142      valid_state(S),
143      write(S), write(' is valid.'), nl.
144      test__vs(S) :-
145      write(S), write(' is invalid.'), nl.
```

Unit Test Demo:

```
1 ?- consult('toh.pro').
true.

2 ?- test__valid_state.
Testing: valid_state
[[l,t,s,m,h],[],[]] is invalid.
[[t,s,m,l,h],[],[]] is valid.
[[],[h,t,s,m],[l]] is invalid.
[[],[t,s,m,h],[l]] is valid.
[[],[h],[l,m,s,t]] is invalid.
[[],[h],[t,s,m,l]] is valid.
true
```

Task 6: Defining the write_sequence predicate

Predicate:

```
write_sequence([]).

write_sequence([H|T]) :-
    sequence(H,A),
    write(A),nl,write_sequence(T).

sequence(m12,This) :-
    This = 'Transfer a disk from tower 1 to tower 2.'.

sequence(m13,This) :-
    This = 'Transfer a disk from tower 1 to tower 3.'.

sequence(m21,This) :-
    This = 'Transfer a disk from tower 2 to tower 1.'.

sequence(m23,This) :-
    This = 'Transfer a disk from tower 2 to tower 3.'.

sequence(m31,This) :-
    This = 'Transfer a disk from tower 3 to tower 1.'.

sequence(m32,This) :-
    This = 'Transfer a disk from tower 3 to tower 2.'.
```

Unit Test Code:

```
test__write_sequence :-
    write('First test of write_sequence ...'), nl,
    write_sequence([m31,m12,m13,m21]),
    write('Second test of write_sequence ...'), nl,
    write_sequence([m13,m12,m32,m13,m21,m23,m13]).
```

Unit Test Demo:

```
1 ?- consult('toh.pro').
true.

2 ?- test__write_sequence.
First test of write_sequence ...
Transfer a disk from tower 3 to tower 1.
Transfer a disk from tower 1 to tower 2.
Transfer a disk from tower 1 to tower 3.
Transfer a disk from tower 2 to tower 1.
Second test of write_sequence ...
Transfer a disk from tower 1 to tower 3.
Transfer a disk from tower 1 to tower 2.
Transfer a disk from tower 3 to tower 2.
Transfer a disk from tower 1 to tower 3.
Transfer a disk from tower 2 to tower 1.
Transfer a disk from tower 2 to tower 3.
Transfer a disk from tower 1 to tower 3.
true.
```

Task 7: Run the program to solve the 3 disk problem

Intermediate with English Output:

```
Microsoft Windows [Version 10.0.22000.613]
(c) Microsoft Corporation. All rights reserved.

C:\Users\zchbo>swipl
Welcome to SWI-Prolog (threaded, 64 bits, version 8.4.2)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

1 ?- cd
.
true.

2 ?- consult('C:/Users/zchbo/Desktop/344 Programming Languages/Prolog/toh.pro').
true.

3 ?- solve.
PathSoFar = [[[s,m,l],[],[]]]
Move = m12
NextState = [[m,l],[s],[]]
PathSoFar = [[[s,m,l],[],[]],[[m,l],[s],[]]]
Move = m12
NextState = [[l],[m,s],[]]
Move = m13
NextState = [[l],[s],[m]]
PathSoFar = [[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]]]
Move = m12
NextState = [[],[l,s],[m]]
Move = m13
NextState = [[],[s],[l,m]]
Move = m21
NextState = [[s,l],[],[m]]
PathSoFar = [[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]]]
Move = m12
NextState = [[l],[s],[m]]
Move = m13
NextState = [[l],[],[s,m]]
PathSoFar = [[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]]]
Move = m12
NextState = [[],[l],[s,m]]
PathSoFar = [[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]]]
Move = m21
```

NextState = [[l],[],[s,m]]

Move = m23

NextState = [[],[],[l,s,m]]

Move = m31

NextState = [[s],[l],[m]]

PathSoFar = [[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]]]

Move = m12

NextState = [[],[s,l],[m]]

PathSoFar = [[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]]]

Move = m21

NextState = [[s],[l],[m]]

Move = m23

NextState = [[],[l],[s,m]]

Move = m31

NextState = [[m],[s,l],[]]

PathSoFar = [[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]]]

Move = m12

NextState = [[],[m,s,l],[]]

Move = m13

NextState = [[],[s,l],[m]]

Move = m21

NextState = [[s,m],[l],[]]

PathSoFar = [[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]]]

Move = m12

NextState = [[m],[s,l],[]]

Move = m13

NextState = [[m],[l],[s]]

PathSoFar = [[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]]]

Move = m12

NextState = [[],[m,l],[s]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]]]

Move = m21

NextState = [[m],[l],[s]]

Move = m23

NextState = [[],[l],[m,s]]

Move = m31

NextState = [[s],[m,l],[]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[s],[m,l],[]]]

Move = m12

NextState = [[],[s,m,l],[]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[s],[m,l],[]],[[],[s,m,l],[]]]

Move = m21

NextState = [[s],[m,l],[]]

Move = m23

NextState = [[],[m,l],[s]]

Move = m13

NextState = [[],[m,l],[s]]

Move = m21

NextState = [[m,s],[l],[]]

Move = m23

NextState = [[s],[l],[m]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[s],[m,l],[]]
]

Move = m12

NextState = [[],[s,m,l],[]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[s],[m,l],[]]
,[[],[s,m,l],[]]]

Move = m21

NextState = [[s],[m,l],[]]

Move = m23

NextState = [[],[m,l],[s]]

Move = m13

NextState = [[],[m,l],[s]]

Move = m21

NextState = [[m,s],[l],[]]

Move = m23

NextState = [[s],[l],[m]]

Move = m32

NextState = [[],[s,m,l],[]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[],[s,m,l],[]
]]

Move = m21

NextState = [[s],[m,l],[]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[],[s,m,l],[]
],[[s],[m,l],[]]]

Move = m12

NextState = [[],[s,m,l],[]]

Move = m13

NextState = [[],[m,l],[s]]

Move = m21

NextState = [[m,s],[l],[]]

Move = m23

NextState = [[s],[l],[m]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[],[s,m,l],[]
],[[s],[m,l],[]]]

Move = m12

NextState = [[],[s,m,l],[]]

Move = m13

NextState = [[],[m,l],[s]]

Move = m21

NextState = [[m,s],[l],[]]

Move = m23

NextState = [[s],[l],[m]]

Move = m23

NextState = [[],[m,l],[s]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]]]

Move = m21

NextState = [[m],[l],[s]]

Move = m23

NextState = [[],[l],[m,s]]

Move = m31

NextState = [[s],[m,l],[]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[s],[m,l],[]]
]

Move = m12

NextState = [[],[s,m,l],[]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[s],[m,l],[]]
,[[],[s,m,l],[]]]

Move = m21

NextState = [[s],[m,l],[]]

Move = m23

NextState = [[],[m,l],[s]]

Move = m13

NextState = [[],[m,l],[s]]

Move = m21

NextState = [[m,s],[l],[]]

Move = m23

NextState = [[s],[l],[m]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[s],[m,l],[]]
]

Move = m12

NextState = [[],[s,m,l],[]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[s],[m,l],[]]
,[[],[s,m,l],[]]]

Move = m21

NextState = [[s],[m,l],[]]

Move = m23

NextState = [[],[m,l],[s]]

Move = m13

NextState = [[],[m,l],[s]]

Move = m21

NextState = [[m,s],[l],[]]

Move = m23

NextState = [[s],[l],[m]]

Move = m32

NextState = [[],[s,m,l],[]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[],[s,m,l],[]]]

Move = m21

NextState = [[s],[m,l],[]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[],[s,m,l],[]],[[s],[m,l],[]]]

Move = m12

NextState = [[],[s,m,l],[]]

Move = m13

NextState = [[],[m,l],[s]]

Move = m21

NextState = [[m,s],[l],[]]

Move = m23

NextState = [[s],[l],[m]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[],[s,m,l],[]],[[s],[m,l],[]]]

Move = m12

NextState = [[],[s,m,l],[]]

Move = m13

NextState = [[],[m,l],[s]]

Move = m21

NextState = [[m,s],[l],[]]

Move = m23

NextState = [[s],[l],[m]]

Move = m23

NextState = [[],[m,l],[s]]

Move = m13

NextState = [[],[l],[m,s]]

Move = m21

NextState = [[l,m],[],[s]]

Move = m23

NextState = [[m],[],[l,s]]

Move = m31

NextState = [[s,m],[l],[]]

Move = m32

NextState = [[m],[s,l],[]]

PathSoFar = [[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]]]

Move = m12

NextState = [[],[m,l],[s]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]]]

Move = m21

NextState = [[m],[l],[s]]

Move = m23

NextState = [[],[l],[m,s]]

Move = m31

NextState = [[s],[m,l],[]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[s],[m,l],[]]
]

Move = m12

NextState = [[],[s,m,l],[]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[s],[m,l],[]]
,[[],[s,m,l],[]]]

Move = m21

NextState = [[s],[m,l],[]]

Move = m23

NextState = [[],[m,l],[s]]

Move = m13

NextState = [[],[m,l],[s]]

Move = m21

NextState = [[m,s],[l],[]]

Move = m23

NextState = [[s],[l],[m]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[s],[m,l],[]]
]

Move = m12

NextState = [[],[s,m,l],[]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[s],[m,l],[]]
,[[],[s,m,l],[]]]

Move = m21

NextState = [[s],[m,l],[]]

Move = m23

NextState = [[],[m,l],[s]]

Move = m13

NextState = [[],[m,l],[s]]

Move = m21

NextState = [[m,s],[l],[]]

Move = m23

NextState = [[s],[l],[m]]

Move = m32

NextState = [[],[s,m,l],[]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[],[s,m,l],[]
]]

Move = m21

NextState = [[s],[m,l],[]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[],[s,m,l],[]
],[[s],[m,l],[]]]

Move = m12

NextState = [[],[s,m,l],[]]

Move = m13

NextState = [[],[m,l],[s]]

Move = m21

NextState = [[m,s],[l],[]]

Move = m23

NextState = [[s],[l],[m]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[],[s,m,l],[]],[[s],[m,l],[]]]

Move = m12

NextState = [[],[s,m,l],[]]

Move = m13

NextState = [[],[m,l],[s]]

Move = m21

NextState = [[m,s],[l],[]]

Move = m23

NextState = [[s],[l],[m]]

Move = m23

NextState = [[],[m,l],[s]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]]]

Move = m21

NextState = [[m],[l],[s]]

Move = m23

NextState = [[],[l],[m,s]]

Move = m31

NextState = [[s],[m,l],[]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[s],[m,l],[]]]

Move = m12

NextState = [[],[s,m,l],[]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[s],[m,l],[]],[[],[s,m,l],[]]]

Move = m21

NextState = [[s],[m,l],[]]

Move = m23

NextState = [[],[m,l],[s]]

Move = m13

NextState = [[],[m,l],[s]]

Move = m21

NextState = [[m,s],[l],[]]

Move = m23

NextState = [[s],[l],[m]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[s],[m,l],[]]]

Move = m12

NextState = [[],[s,m,l],[]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[s],[m,l],[]],[[],[s,m,l],[]]]

Move = m21

NextState = [[s],[m,l],[]]

Move = m23

NextState = [[],[m,l],[s]]

Move = m13

NextState = [[],[m,l],[s]]

Move = m21

NextState = [[m,s],[l],[]]

Move = m23

NextState = [[s],[l],[m]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[s],[m,l],[]],[[],[s,m,l],[]]]

Move = m12

NextState = [[],[s,m,l],[]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[s],[m,l],[]],[[],[s,m,l],[]]]

Move = m21

NextState = [[s],[m,l],[]]

Move = m23

NextState = [[],[m,l],[s]]

Move = m13

NextState = [[],[m,l],[s]]

Move = m21

NextState = [[m,s],[l],[]]

Move = m23

NextState = [[s],[l],[m]]

Move = m32

NextState = [[],[s,m,l],[]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[],[s,m,l],[]]]

Move = m21

NextState = [[s],[m,l],[]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[],[s,m,l],[]],[[s],[m,l],[]]]

Move = m12

NextState = [[],[s,m,l],[]]

Move = m13

NextState = [[],[m,l],[s]]

Move = m21

NextState = [[m,s],[l],[]]

Move = m23

NextState = [[s],[l],[m]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[],[s,m,l],[]],[[s],[m,l],[]]]

Move = m12

NextState = [[],[s,m,l],[]]

Move = m13

NextState = [[],[m,l],[s]]

Move = m21

NextState = [[m,s],[l],[]]

Move = m23

NextState = [[s],[l],[m]]

Move = m23

NextState = [[],[m,l],[s]]

Move = m13

NextState = [[],[l],[m,s]]

Move = m21

NextState = [[l,m],[],[s]]

Move = m23

NextState = [[m],[],[l,s]]

Move = m31

NextState = [[s,m],[l],[]]

Move = m32

NextState = [[m],[s,l],[]]

PathSoFar = [[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]]]

Move = m12

NextState = [[],[m,l],[s]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]]]

Move = m21

NextState = [[m],[l],[s]]

Move = m23

NextState = [[],[l],[m,s]]

Move = m31

NextState = [[s],[m,l],[]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[s],[m,l],[]]]

Move = m12

NextState = [[],[s,m,l],[]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[s],[m,l],[]],[[],[s,m,l],[]]]

Move = m21

NextState = [[s],[m,l],[]]

Move = m23

NextState = [[],[m,l],[s]]

Move = m13

NextState = [[],[m,l],[s]]

Move = m21

NextState = [[m,s],[l],[]]

Move = m23

NextState = [[s],[l],[m]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[s],[m,l],[]]]

Move = m12

NextState = [[],[s,m,l],[]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[s],[m,l],[]],[[],[s,m,l],[]]]

Move = m21

NextState = [[s],[m,l],[]]

Move = m23

NextState = [[],[m,l],[s]]

Move = m13

NextState = [[],[m,l],[s]]

Move = m21

NextState = [[m,s],[l],[]]

Move = m23

NextState = [[s],[l],[m]]

Move = m32

NextState = [[],[s,m,l],[]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[],[s,m,l],[]]]

Move = m21

NextState = [[s],[m,l],[]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[],[s,m,l],[]],[[s],[m,l],[]]]

Move = m12

NextState = [[],[s,m,l],[]]

Move = m13

NextState = [[],[m,l],[s]]

Move = m21

NextState = [[m,s],[l],[]]

Move = m23

NextState = [[s],[l],[m]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[],[s,m,l],[]],[[s],[m,l],[]]]

Move = m12

NextState = [[],[s,m,l],[]]

Move = m13

NextState = [[],[m,l],[s]]

Move = m21

NextState = [[m,s],[l],[]]

Move = m23

NextState = [[s],[l],[m]]

Move = m23

NextState = [[],[m,l],[s]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]]]

Move = m21

NextState = [[m],[l],[s]]

Move = m23

NextState = [[],[l],[m,s]]

Move = m31

NextState = [[s],[m,l],[]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[s],[m,l],[]]]

Move = m12

NextState = [[],[s,m,l],[]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[s],[m,l],[]],[[],[s,m,l],[]]]

Move = m21

NextState = [[s],[m,l],[]]

Move = m23

NextState = [[],[m,l],[s]]

Move = m13

NextState = [[],[m,l],[s]]

Move = m21

NextState = [[m,s],[l],[]]

Move = m23

NextState = [[s],[l],[m]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[s],[m,l],[]]
]

Move = m12

NextState = [[],[s,m,l],[]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[s],[m,l],[]]
,[[],[s,m,l],[]]]

Move = m21

NextState = [[s],[m,l],[]]

Move = m23

NextState = [[],[m,l],[s]]

Move = m13

NextState = [[],[m,l],[s]]

Move = m21

NextState = [[m,s],[l],[]]

Move = m23

NextState = [[s],[l],[m]]

Move = m32

NextState = [[],[s,m,l],[]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[],[s,m,l],[]
]]

Move = m21

NextState = [[s],[m,l],[]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[],[s,m,l],[]
],[[s],[m,l],[]]]

Move = m12

NextState = [[],[s,m,l],[]]

Move = m13

NextState = [[],[m,l],[s]]

Move = m21

NextState = [[m,s],[l],[]]

Move = m23

NextState = [[s],[l],[m]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[],[s,m,l],[]
],[[s],[m,l],[]]]

Move = m12

NextState = [[],[s,m,l],[]]

Move = m13

NextState = [[],[m,l],[s]]

Move = m21

NextState = [[m,s],[l],[]]

Move = m23

NextState = [[s],[l],[m]]

Move = m23

NextState = [[],[m,l],[s]]

Move = m13

NextState = [[],[l],[m,s]]

Move = m21

NextState = [[l,m],[],[s]]

Move = m23

NextState = [[m],[],[l,s]]

Move = m31

NextState = [[s,m],[l],[]]

Move = m32

NextState = [[m],[s,l],[]]

PathSoFar = [[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]]]

Move = m12

NextState = [[],[m,l],[s]]

PathSoFar = [[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]]]

Move = m21

NextState = [[m],[l],[s]]

Move = m23

NextState = [[],[l],[m,s]]

Move = m31

NextState = [[s],[m,l],[]]

PathSoFar = [[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[s],[m,l],[]]]

Move = m12

NextState = [[],[s,m,l],[]]

PathSoFar = [[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[s],[m,l],[]],[[],[s,m,l],[]]]

Move = m21

NextState = [[s],[m,l],[]]

Move = m23

NextState = [[],[m,l],[s]]

Move = m13

NextState = [[],[m,l],[s]]

Move = m21

NextState = [[m,s],[l],[]]

Move = m23

NextState = [[s],[l],[m]]

PathSoFar = [[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[s],[m,l],[]]]

Move = m12

NextState = [[],[s,m,l],[]]

PathSoFar = [[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[s],[m,l],[]],[[],[s,m,l],[]]]

Move = m21

NextState = [[s],[m,l],[]]

Move = m23

NextState = [[],[m,l],[s]]

Move = m13

NextState = [[],[m,l],[s]]

Move = m21

NextState = [[m,s],[l],[]]

Move = m23

NextState = [[s],[l],[m]]

Move = m32

NextState = [[],[s,m,l],[]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[],[s,m,l],[]]]

Move = m21

NextState = [[s],[m,l],[]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[],[s,m,l],[]],[[s],[m,l],[]]]

Move = m12

NextState = [[],[s,m,l],[]]

Move = m13

NextState = [[],[m,l],[s]]

Move = m21

NextState = [[m,s],[l],[]]

Move = m23

NextState = [[s],[l],[m]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[],[s,m,l],[]],[[s],[m,l],[]]]

Move = m12

NextState = [[],[s,m,l],[]]

Move = m13

NextState = [[],[m,l],[s]]

Move = m21

NextState = [[m,s],[l],[]]

Move = m23

NextState = [[s],[l],[m]]

Move = m23

NextState = [[],[m,l],[s]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]]]

Move = m21

NextState = [[m],[l],[s]]

Move = m23

NextState = [[],[l],[m,s]]

Move = m31

NextState = [[s],[m,l],[]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[s],[m,l],[]]]

Move = m12

NextState = [[],[s,m,l],[]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[s],[m,l],[]]
,[[],[s,m,l],[]]]

Move = m21

NextState = [[s],[m,l],[]]

Move = m23

NextState = [[],[m,l],[s]]

Move = m13

NextState = [[],[m,l],[s]]

Move = m21

NextState = [[m,s],[l],[]]

Move = m23

NextState = [[s],[l],[m]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[s],[m,l],[]]
]

Move = m12

NextState = [[],[s,m,l],[]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[s],[m,l],[]]
,[[],[s,m,l],[]]]

Move = m21

NextState = [[s],[m,l],[]]

Move = m23

NextState = [[],[m,l],[s]]

Move = m13

NextState = [[],[m,l],[s]]

Move = m21

NextState = [[m,s],[l],[]]

Move = m23

NextState = [[s],[l],[m]]

Move = m32

NextState = [[],[s,m,l],[]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[],[s,m,l],[]
]]

Move = m21

NextState = [[s],[m,l],[]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[],[s,m,l],[]
],[[s],[m,l],[]]]

Move = m12

NextState = [[],[s,m,l],[]]

Move = m13

NextState = [[],[m,l],[s]]

Move = m21

NextState = [[m,s],[l],[]]

Move = m23

NextState = [[s],[l],[m]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[],[s,m,l],[]],[[s],[m,l],[]]]

Move = m12

NextState = [[],[s,m,l],[]]

Move = m13

NextState = [[],[m,l],[s]]

Move = m21

NextState = [[m,s],[l],[]]

Move = m23

NextState = [[s],[l],[m]]

Move = m23

NextState = [[],[m,l],[s]]

Move = m13

NextState = [[],[l],[m,s]]

Move = m21

NextState = [[l,m],[],[s]]

Move = m23

NextState = [[m],[],[l,s]]

Move = m31

NextState = [[s,m],[l],[]]

Move = m32

NextState = [[m],[s,l],[]]

PathSoFar = [[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]]]

Move = m12

NextState = [[],[m,l],[s]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]]]

Move = m21

NextState = [[m],[l],[s]]

Move = m23

NextState = [[],[l],[m,s]]

Move = m31

NextState = [[s],[m,l],[]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[s],[m,l],[]]]

Move = m12

NextState = [[],[s,m,l],[]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[s],[m,l],[]],[[],[s,m,l],[]]]

Move = m21

NextState = [[s],[m,l],[]]

Move = m23

NextState = [[],[m,l],[s]]

Move = m13

NextState = [[],[m,l],[s]]

Move = m21

NextState = [[m,s],[l],[]]

Move = m23

NextState = [[s],[l],[m]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[s],[m,l],[]]
]

Move = m12

NextState = [[],[s,m,l],[]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[s],[m,l],[]]
,[[],[s,m,l],[]]]

Move = m21

NextState = [[s],[m,l],[]]

Move = m23

NextState = [[],[m,l],[s]]

Move = m13

NextState = [[],[m,l],[s]]

Move = m21

NextState = [[m,s],[l],[]]

Move = m23

NextState = [[s],[l],[m]]

Move = m32

NextState = [[],[s,m,l],[]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[],[s,m,l],[]
]]

Move = m21

NextState = [[s],[m,l],[]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[],[s,m,l],[]
],[[s],[m,l],[]]]

Move = m12

NextState = [[],[s,m,l],[]]

Move = m13

NextState = [[],[m,l],[s]]

Move = m21

NextState = [[m,s],[l],[]]

Move = m23

NextState = [[s],[l],[m]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[],[s,m,l],[]
],[[s],[m,l],[]]]

Move = m12

NextState = [[],[s,m,l],[]]

Move = m13

NextState = [[],[m,l],[s]]

Move = m21

NextState = [[m,s],[l],[]]

Move = m23

NextState = [[s],[l],[m]]

Move = m23

NextState = [[],[m,l],[s]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]]]

Move = m21

NextState = [[m],[l],[s]]

Move = m23

NextState = [[],[l],[m,s]]

Move = m31

NextState = [[s],[m,l],[]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[s],[m,l],[]]]

Move = m12

NextState = [[],[s,m,l],[]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[s],[m,l],[]],[[],[s,m,l],[]]]

Move = m21

NextState = [[s],[m,l],[]]

Move = m23

NextState = [[],[m,l],[s]]

Move = m13

NextState = [[],[m,l],[s]]

Move = m21

NextState = [[m,s],[l],[]]

Move = m23

NextState = [[s],[l],[m]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[s],[m,l],[]]]

Move = m12

NextState = [[],[s,m,l],[]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[s],[m,l],[]],[[],[s,m,l],[]]]

Move = m21

NextState = [[s],[m,l],[]]

Move = m23

NextState = [[],[m,l],[s]]

Move = m13

NextState = [[],[m,l],[s]]

Move = m21

NextState = [[m,s],[l],[]]

Move = m23

NextState = [[s],[l],[m]]

Move = m32

NextState = [[],[s,m,l],[]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[],[s,m,l],[]]]

Move = m21

NextState = [[s],[m,l],[]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[],[s,m,l],[]],[[s],[m,l],[]]]

Move = m12

NextState = [[],[s,m,l],[]]

Move = m13

NextState = [[],[m,l],[s]]

Move = m21

NextState = [[m,s],[l],[]]

Move = m23

NextState = [[s],[l],[m]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[],[s,m,l],[]],[[s],[m,l],[]]]

Move = m12

NextState = [[],[s,m,l],[]]

Move = m13

NextState = [[],[m,l],[s]]

Move = m21

NextState = [[m,s],[l],[]]

Move = m23

NextState = [[s],[l],[m]]

Move = m23

NextState = [[],[m,l],[s]]

Move = m13

NextState = [[],[l],[m,s]]

Move = m21

NextState = [[l,m],[],[s]]

Move = m23

NextState = [[m],[],[l,s]]

Move = m31

NextState = [[s,m],[l],[]]

Move = m32

NextState = [[m],[s,l],[]]

PathSoFar = [[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]]]

Move = m12

NextState = [[],[m,l],[s]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]]]

Move = m21

NextState = [[m],[l],[s]]

Move = m23

NextState = [[],[l],[m,s]]

Move = m31

NextState = [[s],[m,l],[]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[s],[m,l],[]]]

Move = m12

NextState = [[],[s,m,l],[]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[s],[m,l],[]]
,[[],[s,m,l],[]]]

Move = m21

NextState = [[s],[m,l],[]]

Move = m23

NextState = [[],[m,l],[s]]

Move = m13

NextState = [[],[m,l],[s]]

Move = m21

NextState = [[m,s],[l],[]]

Move = m23

NextState = [[s],[l],[m]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[s],[m,l],[]]
]

Move = m12

NextState = [[],[s,m,l],[]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[s],[m,l],[]]
,[[],[s,m,l],[]]]

Move = m21

NextState = [[s],[m,l],[]]

Move = m23

NextState = [[],[m,l],[s]]

Move = m13

NextState = [[],[m,l],[s]]

Move = m21

NextState = [[m,s],[l],[]]

Move = m23

NextState = [[s],[l],[m]]

Move = m32

NextState = [[],[s,m,l],[]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[],[s,m,l],[]
]]

Move = m21

NextState = [[s],[m,l],[]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[],[s,m,l],[]
],[[s],[m,l],[]]]

Move = m12

NextState = [[],[s,m,l],[]]

Move = m13

NextState = [[],[m,l],[s]]

Move = m21

NextState = [[m,s],[l],[]]

Move = m23

NextState = [[s],[l],[m]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[],[s,m,l],[]
],[[s],[m,l],[]]]

Move = m12

NextState = [[],[s,m,l],[]]

Move = m13

NextState = [[],[m,l],[s]]

Move = m21

NextState = [[m,s],[l],[]]

Move = m23

NextState = [[s],[l],[m]]

Move = m23

NextState = [[],[m,l],[s]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]]]

Move = m21

NextState = [[m],[l],[s]]

Move = m23

NextState = [[],[l],[m,s]]

Move = m31

NextState = [[s],[m,l],[]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[s],[m,l],[]]]

Move = m12

NextState = [[],[s,m,l],[]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[s],[m,l],[]],[[],[s,m,l],[]]]

Move = m21

NextState = [[s],[m,l],[]]

Move = m23

NextState = [[],[m,l],[s]]

Move = m13

NextState = [[],[m,l],[s]]

Move = m21

NextState = [[m,s],[l],[]]

Move = m23

NextState = [[s],[l],[m]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[s],[m,l],[]]]

Move = m12

NextState = [[],[s,m,l],[]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[s],[m,l],[]],[[],[s,m,l],[]]]

Move = m21

NextState = [[s],[m,l],[]]

Move = m23

NextState = [[],[m,l],[s]]

Move = m13

NextState = [[],[m,l],[s]]

Move = m21

NextState = [[m,s],[l],[]]

Move = m23

NextState = [[s],[l],[m]]

Move = m32

NextState = [[],[s,m,l],[]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[],[s,m,l],[]]]

Move = m21

NextState = [[s],[m,l],[]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[],[s,m,l],[]],[[s],[m,l],[]]]

Move = m12

NextState = [[],[s,m,l],[]]

Move = m13

NextState = [[],[m,l],[s]]

Move = m21

NextState = [[m,s],[l],[]]

Move = m23

NextState = [[s],[l],[m]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[],[s,m,l],[]],[[s],[m,l],[]]]

Move = m12

NextState = [[],[s,m,l],[]]

Move = m13

NextState = [[],[m,l],[s]]

Move = m21

NextState = [[m,s],[l],[]]

Move = m23

NextState = [[s],[l],[m]]

Move = m23

NextState = [[],[m,l],[s]]

Move = m13

NextState = [[],[l],[m,s]]

Move = m21

NextState = [[l,m],[],[s]]

Move = m23

NextState = [[m],[],[l,s]]

Move = m31

NextState = [[s,m],[l],[]]

Move = m32

NextState = [[m],[s,l],[]]

PathSoFar = [[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]]]

Move = m12

NextState = [[],[m,l],[s]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]]]

Move = m21

NextState = [[m],[l],[s]]

Move = m23

NextState = [[],[l],[m,s]]

Move = m31

NextState = [[s],[m,l],[]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[s],[m,l],[]]
]

Move = m12

NextState = [[],[s,m,l],[]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[s],[m,l],[]]
,[[],[s,m,l],[]]]

Move = m21

NextState = [[s],[m,l],[]]

Move = m23

NextState = [[],[m,l],[s]]

Move = m13

NextState = [[],[m,l],[s]]

Move = m21

NextState = [[m,s],[l],[]]

Move = m23

NextState = [[s],[l],[m]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[s],[m,l],[]]
]

Move = m12

NextState = [[],[s,m,l],[]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[s],[m,l],[]]
,[[],[s,m,l],[]]]

Move = m21

NextState = [[s],[m,l],[]]

Move = m23

NextState = [[],[m,l],[s]]

Move = m13

NextState = [[],[m,l],[s]]

Move = m21

NextState = [[m,s],[l],[]]

Move = m23

NextState = [[s],[l],[m]]

Move = m32

NextState = [[],[s,m,l],[]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[],[s,m,l],[]
]]

Move = m21

NextState = [[s],[m,l],[]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[],[s,m,l],[]
],[[s],[m,l],[]]]

Move = m12

NextState = [[],[s,m,l],[]]

Move = m13

NextState = [[],[m,l],[s]]

Move = m21

NextState = [[m,s],[l],[]]

Move = m23

NextState = [[s],[l],[m]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[],[s,m,l],[]],[[s],[m,l],[]]]

Move = m12

NextState = [[],[s,m,l],[]]

Move = m13

NextState = [[],[m,l],[s]]

Move = m21

NextState = [[m,s],[l],[]]

Move = m23

NextState = [[s],[l],[m]]

Move = m23

NextState = [[],[m,l],[s]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]]]

Move = m21

NextState = [[m],[l],[s]]

Move = m23

NextState = [[],[l],[m,s]]

Move = m31

NextState = [[s],[m,l],[]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[s],[m,l],[]]]

Move = m12

NextState = [[],[s,m,l],[]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[s],[m,l],[]],[[],[s,m,l],[]]]

Move = m21

NextState = [[s],[m,l],[]]

Move = m23

NextState = [[],[m,l],[s]]

Move = m13

NextState = [[],[m,l],[s]]

Move = m21

NextState = [[m,s],[l],[]]

Move = m23

NextState = [[s],[l],[m]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[s],[m,l],[]]]

Move = m12

NextState = [[],[s,m,l],[]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[s],[m,l],[]],[[],[s,m,l],[]]]

Move = m21

NextState = [[s],[m,l],[]]

Move = m23

NextState = [[],[m,l],[s]]

Move = m13

NextState = [[],[m,l],[s]]

Move = m21

NextState = [[m,s],[l],[]]

Move = m23

NextState = [[s],[l],[m]]

Move = m32

NextState = [[],[s,m,l],[]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[],[s,m,l],[]]]

Move = m21

NextState = [[s],[m,l],[]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[],[s,m,l],[]],[[s],[m,l],[]]]

Move = m12

NextState = [[],[s,m,l],[]]

Move = m13

NextState = [[],[m,l],[s]]

Move = m21

NextState = [[m,s],[l],[]]

Move = m23

NextState = [[s],[l],[m]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[],[s,m,l],[]],[[s],[m,l],[]]]

Move = m12

NextState = [[],[s,m,l],[]]

Move = m13

NextState = [[],[m,l],[s]]

Move = m21

NextState = [[m,s],[l],[]]

Move = m23

NextState = [[s],[l],[m]]

Move = m23

NextState = [[],[m,l],[s]]

Move = m13

NextState = [[],[l],[m,s]]

Move = m21

NextState = [[l,m],[],[s]]

Move = m23

NextState = [[m],[],[l,s]]

Move = m31

NextState = [[s,m],[l],[]]

Move = m32

NextState = [[m],[s,l],[]]

PathSoFar = [[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]]]

Move = m12

NextState = [[],[m,l],[s]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]]]

Move = m21

NextState = [[m],[l],[s]]

Move = m23

NextState = [[],[l],[m,s]]

Move = m31

NextState = [[s],[m,l],[]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[s],[m,l],[]]]

Move = m12

NextState = [[],[s,m,l],[]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[s],[m,l],[]],[[],[s,m,l],[]]]

Move = m21

NextState = [[s],[m,l],[]]

Move = m23

NextState = [[],[m,l],[s]]

Move = m13

NextState = [[],[m,l],[s]]

Move = m21

NextState = [[m,s],[l],[]]

Move = m23

NextState = [[s],[l],[m]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[s],[m,l],[]]]

Move = m12

NextState = [[],[s,m,l],[]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[s],[m,l],[]],[[],[s,m,l],[]]]

Move = m21

NextState = [[s],[m,l],[]]

Move = m23

NextState = [[],[m,l],[s]]

Move = m13

NextState = [[],[m,l],[s]]

Move = m21

NextState = [[m,s],[l],[]]

Move = m23

NextState = [[s],[l],[m]]

Move = m32

NextState = [[],[s,m,l],[]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[],[s,m,l],[]]]

Move = m21

NextState = [[s],[m,l],[]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[],[s,m,l],[]],[[s],[m,l],[]]]

Move = m12

NextState = [[],[s,m,l],[]]

Move = m13

NextState = [[],[m,l],[s]]

Move = m21

NextState = [[m,s],[l],[]]

Move = m23

NextState = [[s],[l],[m]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[],[s,m,l],[]],[[s],[m,l],[]]]

Move = m12

NextState = [[],[s,m,l],[]]

Move = m13

NextState = [[],[m,l],[s]]

Move = m21

NextState = [[m,s],[l],[]]

Move = m23

NextState = [[s],[l],[m]]

Move = m23

NextState = [[],[m,l],[s]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]]]

Move = m21

NextState = [[m],[l],[s]]

Move = m23

NextState = [[],[l],[m,s]]

Move = m31

NextState = [[s],[m,l],[]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[s],[m,l],[]]]

Move = m12

NextState = [[],[s,m,l],[]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[s],[m,l],[]],[[],[s,m,l],[]]]

Move = m21

NextState = [[s],[m,l],[]]

Move = m23

NextState = [[],[m,l],[s]]

Move = m13

NextState = [[],[m,l],[s]]

Move = m21

NextState = [[m,s],[l],[]]

Move = m23

NextState = [[s],[l],[m]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[s],[m,l],[]]
]

Move = m12

NextState = [[],[s,m,l],[]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[s],[m,l],[]]
,[[],[s,m,l],[]]]

Move = m21

NextState = [[s],[m,l],[]]

Move = m23

NextState = [[],[m,l],[s]]

Move = m13

NextState = [[],[m,l],[s]]

Move = m21

NextState = [[m,s],[l],[]]

Move = m23

NextState = [[s],[l],[m]]

Move = m32

NextState = [[],[s,m,l],[]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[],[s,m,l],[]
]]

Move = m21

NextState = [[s],[m,l],[]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[],[s,m,l],[]
],[[s],[m,l],[]]]

Move = m12

NextState = [[],[s,m,l],[]]

Move = m13

NextState = [[],[m,l],[s]]

Move = m21

NextState = [[m,s],[l],[]]

Move = m23

NextState = [[s],[l],[m]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[m],[l],[s]],[[],[m,l],[s]],[[],[s,m,l],[]
],[[s],[m,l],[]]]

Move = m12

NextState = [[],[s,m,l],[]]

Move = m13

NextState = [[],[m,l],[s]]

Move = m21

NextState = [[m,s],[l],[]]

Move = m23

NextState = [[s],[l],[m]]

Move = m23

NextState = [[],[m,l],[s]]

Move = m13

NextState = [[],[l],[m,s]]

Move = m21

NextState = [[l,m],[],[s]]

Move = m23

NextState = [[m],[],[l,s]]

Move = m31

NextState = [[s,m],[l],[]]

Move = m32

NextState = [[m],[s,l],[]]

Move = m21

NextState = [[l,s,m],[],[]]

Move = m23

NextState = [[s,m],[],[l]]

PathSoFar = [[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[s,m],[],[l]]]

Move = m12

NextState = [[m],[s],[l]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[s,m],[],[l]],[[m],[s],[l]]]

Move = m12

NextState = [[],[m,s],[l]]

Move = m13

NextState = [[],[s],[m,l]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[s,m],[],[l]],[[m],[s],[l]],[[],[s],[m,l]]]

Move = m21

NextState = [[s],[],[m,l]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[s,m],[],[l]],[[m],[s],[l]],[[],[s],[m,l]],[[s],[],[m,l]]]

Move = m12

NextState = [[],[s],[m,l]]

Move = m13

NextState = [[],[],[s,m,l]]

PathSoFar =
[[[s,m,l],[],[]],[[m,l],[s],[]],[[l],[s],[m]],[[s,l],[],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[],[s,l],[m]],[[m],[s,l],[]],[[s,m],[l],[]],[[s,m],[],[l]],[[m],[s],[l]],[[],[s],[m,l]],[[s],[],[m,l]],[[],[],[s,m,l]]]

SolutionSoFar = [m12,m13,m21,m13,m12,m31,m12,m31,m21,m23,m12,m13,m21,m13]

Solution ...

Transfer a disk from tower 1 to tower 2.

Transfer a disk from tower 1 to tower 3.

Transfer a disk from tower 2 to tower 1.

Transfer a disk from tower 1 to tower 3.

Transfer a disk from tower 1 to tower 2.
Transfer a disk from tower 3 to tower 1.
Transfer a disk from tower 1 to tower 2.
Transfer a disk from tower 3 to tower 1.
Transfer a disk from tower 2 to tower 1.
Transfer a disk from tower 2 to tower 3.
Transfer a disk from tower 1 to tower 2.
Transfer a disk from tower 1 to tower 3.
Transfer a disk from tower 2 to tower 1.
Transfer a disk from tower 1 to tower 3.

true

Just English Output:

Solution ...

Transfer a disk from tower 1 to tower 2.
Transfer a disk from tower 1 to tower 3.
Transfer a disk from tower 2 to tower 1.
Transfer a disk from tower 1 to tower 3.
Transfer a disk from tower 1 to tower 2.
Transfer a disk from tower 3 to tower 1.
Transfer a disk from tower 1 to tower 2.
Transfer a disk from tower 3 to tower 1.
Transfer a disk from tower 2 to tower 1.
Transfer a disk from tower 2 to tower 3.
Transfer a disk from tower 1 to tower 2.
Transfer a disk from tower 1 to tower 3.
Transfer a disk from tower 2 to tower 1.
Transfer a disk from tower 1 to tower 3.

true

Questions:

1. What was the length of your program's solution to the three disk problem?
   14 Steps.
2. What is the length of the shortest solution to the three disk problem?
   7 steps.
3. How do you account for the discrepancy?
   The program is using trial and error. It does not plan it's next move, instead it makes a move, and then another and so on until it is solved.

Task 8: Run the program to solve the 4 disk problem

Demo:

```
Solution ...

Transfer a disk from tower 1 to tower 2.
Transfer a disk from tower 1 to tower 3.
Transfer a disk from tower 2 to tower 1.
Transfer a disk from tower 1 to tower 3.
Transfer a disk from tower 1 to tower 2.
Transfer a disk from tower 3 to tower 1.
Transfer a disk from tower 1 to tower 2.
Transfer a disk from tower 3 to tower 1.
Transfer a disk from tower 1 to tower 2.
Transfer a disk from tower 1 to tower 3.
Transfer a disk from tower 2 to tower 1.
Transfer a disk from tower 1 to tower 3.
Transfer a disk from tower 2 to tower 1.
Transfer a disk from tower 3 to tower 1.
Transfer a disk from tower 1 to tower 2.
Transfer a disk from tower 1 to tower 3.
Transfer a disk from tower 2 to tower 1.
Transfer a disk from tower 1 to tower 3.

true .
```

Questions:

1. What was the length of your program's solution to the four disk problem?
   18 moves.
2. What is the length of the shortest solution to the four disk problem?
   15 moves.

Task 9: Review your code and archive it

Inspector.pro

```
% ------------------------------------------------------------------------
% ------------------------------------------------------------------------
% --- File: inspectors.pro
% --- Line: Utilities for inspecting memory during program execution
% ------------------------------------------------------------------------
% ------------------------------------------------------------------------
% --- These two can be used to print the value of a variable, labelled
% --- in two ways, and pause for the programmer to check out the
% --- situation. The firstone is generally useful. The second one is
% --- applicable only whenthe value of the veriable is a list, and it
% --- will print the valuein reverse order which is sometimes just
% --- what is desired. The first label generally pertains to a location
% --- in the program. The second label is just the name of the variable
% --- to which the value is bound.
check(Label,Name,Value) :-
    write(Label),
    write(Name),write(' = '),
    write(Value),nl,
    read(_).
    checkr(Label,Name,Value) :-
    write(Label),
    write(Name),write(' = '),
    reverse(Value,RValue),
    write(RValue),nl,
    read(_).
    % ------------------------------------------------------------------------
    % --- These two are like the previously described checking predicates,
    % --- except that they do not do the pause.
    trace(Label,Name,Value) :-
    write(Label),
    write(Name),write(' = '),
    write(Value),nl.
    tracer(Label,Name,Value) :-
    write(Label),
    write(Name),write(' = '),
    reverse(Value,RValue),
    write(RValue),nl.
    % ------------------------------------------------------------------------
    % --- Like trace, but without the extra labelling functionality.
    show(Name,Value) :-
    write(Name),write(' = '),
    write(Value),nl.
    showr(Name,Value) :-
    write(Name),write(' = '),
    reverse(Value,RValue),
    write(RValue),nl.
```

Toh.pro

```
% ------------------------------------------------------------------------
% ------------------------------------------------------------------------
% --- File: towers_of_hanoi.pro
% --- Line: Program to solve the Towers of Hanoi problem
% ------------------------------------------------------------------------
:- consult('inspector.pro').
% ------------------------------------------------------------------------
% --- make_move(S,T,SSO) :: Make a move from state S to state T by SSO
make_move(TowersBeforeMove,TowersAfterMove,m12) :-
m12(TowersBeforeMove,TowersAfterMove).
make_move(TowersBeforeMove,TowersAfterMove,m13) :-
m13(TowersBeforeMove,TowersAfterMove).
make_move(TowersBeforeMove,TowersAfterMove,m21) :-
m21(TowersBeforeMove,TowersAfterMove).
make_move(TowersBeforeMove,TowersAfterMove,m23) :-
m23(TowersBeforeMove,TowersAfterMove).
make_move(TowersBeforeMove,TowersAfterMove,m31) :-
m31(TowersBeforeMove,TowersAfterMove).
make_move(TowersBeforeMove,TowersAfterMove,m32) :-
m32(TowersBeforeMove,TowersAfterMove).
% ------------------------------------------------------------------------
% --- valid_state(S) :: S is a valid state
% ------------------------------------------------------------------------
% --- solve(Start,Solution) :: succeeds if Solution represents a path
% --- from the start state to the goal state.
solve :-
extend_path([[[s,m,l,h],[],[]]],[],Solution),
write_solution(Solution).
extend_path(PathSoFar,SolutionSoFar,Solution) :-
PathSoFar = [[[],[],[s,m,l,h]]|_],
showr('PathSoFar',PathSoFar),
showr('SolutionSoFar',SolutionSoFar),
Solution = SolutionSoFar.
extend_path(PathSoFar,SolutionSoFar,Solution) :-
PathSoFar = [CurrentState|_],
showr('PathSoFar',PathSoFar),
make_move(CurrentState,NextState,Move),
show('Move',Move),
show('NextState',NextState),
not(member(NextState,PathSoFar)),
valid_state(NextState),
Path = [NextState|PathSoFar],
Soln = [Move|SolutionSoFar],
extend_path(Path,Soln,Solution).
% ------------------------------------------------------------------------
% --- write_sequence_reversed(S) :: Write the sequence, given by S,
% --- expanding the tokens into meaningful strings.
write_solution(S) :-
```

```
49  nl, write('Solution ...'), nl, nl,
50  reverse(S,R),
51  write_sequence(R),nl.
52  % ----------------------------------------------------------------------
53  % --- Unit test programs
54
55  m12([Tower1Before,Tower2Before,Tower3],[Tower1After,Tower2After,Tower3]) :-
56      Tower1Before = [H|T],
57      Tower1After = T,
58      Tower2Before = L,
59      Tower2After = [H|L].
60
61  m13([Tower1Before,Tower2,Tower3Before],[Tower1After,Tower2,Tower3After]) :-
62      Tower1Before = [H|T],
63      Tower1After = T,
64      Tower3Before = L,
65      Tower3After = [H|L].
66
67  m21([Tower1Before,Tower2Before,Tower3],[Tower1After,Tower2After,Tower3]) :-
68      Tower2Before = [H|T],
69      Tower2After = T,
70      Tower1Before = L,
71      Tower1After = [H|L].
72
73  m23([Tower1,Tower2Before,Tower3Before],[Tower1,Tower2After,Tower3After]) :-
74      Tower2Before = [H|T],
75      Tower2After = T,
76      Tower3Before = L,
77      Tower3After = [H|L].
78
79  m31([Tower1Before,Tower2,Tower3Before],[Tower1After,Tower2,Tower3After]) :-
80      Tower3Before = [H|T],
81      Tower3After = T,
82      Tower1Before = L,
83      Tower1After = [H|L].
84
85  m32([Tower1,Tower2Before,Tower3Before],[Tower1,Tower2After,Tower3After]) :-
86      Tower3Before = [H|T],
87      Tower3After = T,
88      Tower2Before = L,
89      Tower2After = [H|L].
90
91  test__m12 :-
92      write('Testing: move_m12\n'),
93      TowersBefore = [[t,s,m,l,h],[],[]],
94      trace('','TowersBefore',TowersBefore),
95      m12(TowersBefore,TowersAfter),
```

```
96      trace('','TowersAfter',TowersAfter).
97
98  test__m13 :-
99      write('Testing: move_m13\n'),
100     TowersBefore = [[t,s,m,l,h],[],[]],
101     trace('','TowersBefore',TowersBefore),
102     m13(TowersBefore,TowersAfter),
103     trace('','TowersAfter',TowersAfter).
104
105 test__m21 :-
106     write('Testing: move_m21\n'),
107     TowersBefore = [[m,l,h],[t,s],[]],
108     trace('','TowersBefore',TowersBefore),
109     m21(TowersBefore,TowersAfter),
110     trace('','TowersAfter',TowersAfter).
111
112 test__m23 :-
113     write('Testing: move_m23\n'),
114     TowersBefore = [[l,h],[t,s],[m]],
115     trace('','TowersBefore',TowersBefore),
116     m23(TowersBefore,TowersAfter),
117     trace('','TowersAfter',TowersAfter).
118
119 test__m31 :-
120     write('Testing: move_m31\n'),
121     TowersBefore = [[m,l,h],[],[t,s]],
122     trace('','TowersBefore',TowersBefore),
123     m31(TowersBefore,TowersAfter),
124     trace('','TowersAfter',TowersAfter).
125
126 test__m32 :-
127     write('Testing: move_m32\n'),
128     TowersBefore = [[l,h],[m],[t,s]],
129     trace('','TowersBefore',TowersBefore),
130     m32(TowersBefore,TowersAfter),
131     trace('','TowersAfter',TowersAfter).
132
133 test__valid_state :-
134     write('Testing: valid_state\n'),
135     test__vs([[l,t,s,m,h],[],[]]),
136     test__vs([[t,s,m,l,h],[],[]]),
137     test__vs([[],[h,t,s,m],[l]]),
138     test__vs([[],[t,s,m,h],[l]]),
139     test__vs([[],[h],[l,m,s,t]]),
140     test__vs([[],[h],[t,s,m,l]]).
141     test__vs(S) :-
142     valid_state(S),
143     write(S), write(' is valid.'), nl.
```

```
144     test__vs(S) :-
145     write(S), write(' is invalid.'), nl.
146
147 valid_state([A|[B|[C]]]) :- state(A), state(B), state(C).
148     state([]).
149     state([t]).
150     state([s]).
151     state([m]).
152     state([l]).
153     state([h]).
154     state([t,s]).
155     state([t,s,m]).
156     state([t,s,m,l]).
157     state([t,s,l]).
158     state([t,s,l,h]).
159     state([t,s,h]).
160     state([t,s,m,h]).
161     state([t,m]).
162     state([t,m,l]).
163     state([t,m,l,h]).
164     state([t,m,h]).
165     state([t,l,h]).
166     state([t,l]).
167     state([t,h]).
168     state([t,s,m,l,h]).
169     state([s]).
170     state([s,m]).
171     state([s,m,l]).
172     state([s,m,l,h]).
173     state([s,l]).
174     state([s,l,h]).
175     state([s,h]).
176     state([s,m,h]).
177     state([m]).
178     state([m,l]).
179     state([m,l,h]).
180     state([m,h]).
181     state([l]).
182     state([l,h]).
183     state([h]).
184
185
186 write_sequence([]).
187
188 write_sequence([H|T]) :-
189     sequence(H,A),
190     write(A) nl write_sequence(T)
```

```
191
192   sequence(m12,This) :-
193       This = 'Transfer a disk from tower 1 to tower 2.'.
194
195   sequence(m13,This) :-
196       This = 'Transfer a disk from tower 1 to tower 3.'.
197
198   sequence(m21,This) :-
199       This = 'Transfer a disk from tower 2 to tower 1.'.
200
201   sequence(m23,This) :-
202       This = 'Transfer a disk from tower 2 to tower 3.'.
203
204   sequence(m31,This) :-
205       This = 'Transfer a disk from tower 3 to tower 1.'.
206
207   sequence(m32,This) :-
208       This = 'Transfer a disk from tower 3 to tower 2.'.
209
210
211
212   test__write_sequence :-
213       write('First test of write_sequence ...'), nl,
214       write_sequence([m31,m12,m13,m21]),
215       write('Second test of write_sequence ...'), nl,
216       write_sequence([m13,m12,m32,m13,m21,m23,m13]).
```