# Missionaries with ItrDFS

Welcome to ItrDFS
Initial State:

 M on left:3
 C on left:3
   M on right:0
   C on right:0
 boat is on the left.

len(OPEN)=1; len(CLOSED)=0; COUNT = 0
OPEN is now:
 M on left:0
 C on left:3
   M on right:3
   C on right:0
 boat is on the right.
,
 M on left:2
 C on left:2
   M on right:1
   C on right:1
 boat is on the right.

len(OPEN)=2; len(CLOSED)=1; COUNT = 1
OPEN is now:
 M on left:2
 C on left:2
   M on right:1
   C on right:1
 boat is on the right.

len(OPEN)=1; len(CLOSED)=2; COUNT = 2
OPEN is now:
 M on left:3
 C on left:2
   M on right:0
   C on right:1
 boat is on the left.

len(OPEN)=1; len(CLOSED)=3; COUNT = 3
OPEN is now:
 M on left:0
 C on left:2
   M on right:3
   C on right:1
 boat is on the right.
,
 M on left:1
 C on left:1

```
   M on right:2
   C on right:2
 boat is on the right.

len(OPEN)=2; len(CLOSED)=4; COUNT = 4
OPEN is now:
 M on left:2
 C on left:2
   M on right:1
   C on right:1
 boat is on the left.
,
 M on left:1
 C on left:1
   M on right:2
   C on right:2
 boat is on the right.

len(OPEN)=2; len(CLOSED)=5; COUNT = 5
OPEN is now:
 M on left:1
 C on left:1
   M on right:2
   C on right:2
 boat is on the right.
,
 M on left:0
 C on left:1
   M on right:3
   C on right:2
 boat is on the right.

len(OPEN)=2; len(CLOSED)=6; COUNT = 6
OPEN is now:
 M on left:3
 C on left:1
   M on right:0
   C on right:2
 boat is on the left.
,
 M on left:0
 C on left:1
   M on right:3
   C on right:2
 boat is on the right.

len(OPEN)=2; len(CLOSED)=7; COUNT = 7
OPEN is now:
 M on left:0
 C on left:1
   M on right:3
   C on right:2
 boat is on the right.
```

len(OPEN)=1; len(CLOSED)=8; COUNT = 8
OPEN is now:
 M on left:1
 C on left:1
   M on right:2
   C on right:2
 boat is on the left.

len(OPEN)=1; len(CLOSED)=9; COUNT = 9
OPEN is now:
 M on left:0
 C on left:0
   M on right:3
   C on right:3
 boat is on the right.

len(OPEN)=1; len(CLOSED)=10; COUNT = 10
Congratulations on successfully guiding the missionaries and cannibals across the river!
Solution path:

 M on left:3
 C on left:3
   M on right:0
   C on right:0
 boat is on the left.


 M on left:2
 C on left:2
   M on right:1
   C on right:1
 boat is on the right.


 M on left:3
 C on left:2
   M on right:0
   C on right:1
 boat is on the left.


 M on left:0
 C on left:2
   M on right:3
   C on right:1
 boat is on the right.


 M on left:2
 C on left:2
   M on right:1
   C on right:1
 boat is on the left.

M on left:1
C on left:1
  M on right:2
  C on right:2
boat is on the right.

M on left:3
C on left:1
  M on right:0
  C on right:2
boat is on the left.

M on left:0
C on left:1
  M on right:3
  C on right:2
boat is on the right.

M on left:1
C on left:1
  M on right:2
  C on right:2
boat is on the left.

M on left:0
C on left:0
  M on right:3
  C on right:3
boat is on the right.

Length of solution path found: 9 edges
10 states expanded.
MAX_OPEN_LENGTH = 2

# Missionaries with BFS

Welcome to BFS
Initial State:

 M on left:3
 C on left:3
   M on right:0
   C on right:0
 boat is on the left.

len(OPEN)=1; len(CLOSED)=0; COUNT = 0
OPEN is now:
 M on left:0
 C on left:3
   M on right:3
   C on right:0
 boat is on the right.
,
 M on left:2
 C on left:2
   M on right:1
   C on right:1
 boat is on the right.

len(OPEN)=2; len(CLOSED)=1; COUNT = 1
OPEN is now:
 M on left:2
 C on left:2
   M on right:1
   C on right:1
 boat is on the right.

len(OPEN)=1; len(CLOSED)=2; COUNT = 2
OPEN is now:
 M on left:3
 C on left:2
   M on right:0
   C on right:1
 boat is on the left.

len(OPEN)=1; len(CLOSED)=3; COUNT = 3
OPEN is now:
 M on left:0
 C on left:2
   M on right:3
   C on right:1
 boat is on the right.
,

```
 M on left:1
 C on left:1
   M on right:2
   C on right:2
 boat is on the right.

len(OPEN)=2; len(CLOSED)=4; COUNT = 4
OPEN is now:
 M on left:1
 C on left:1
   M on right:2
   C on right:2
 boat is on the right.
,
 M on left:2
 C on left:2
   M on right:1
   C on right:1
 boat is on the left.

len(OPEN)=2; len(CLOSED)=5; COUNT = 5
OPEN is now:
 M on left:2
 C on left:2
   M on right:1
   C on right:1
 boat is on the left.
,
 M on left:3
 C on left:1
   M on right:0
   C on right:2
 boat is on the left.

len(OPEN)=2; len(CLOSED)=6; COUNT = 6
OPEN is now:
 M on left:3
 C on left:1
   M on right:0
   C on right:2
 boat is on the left.
,
 M on left:0
 C on left:1
   M on right:3
   C on right:2
 boat is on the right.

len(OPEN)=2; len(CLOSED)=7; COUNT = 7
OPEN is now:
 M on left:0
 C on left:1
   M on right:3
   C on right:2
```

boat is on the right.

len(OPEN)=1; len(CLOSED)=8; COUNT = 8
OPEN is now:
M on left:1
C on left:1
M on right:2
C on right:2
boat is on the left.

len(OPEN)=1; len(CLOSED)=9; COUNT = 9
OPEN is now:
M on left:0
C on left:0
M on right:3
C on right:3
boat is on the right.

len(OPEN)=1; len(CLOSED)=10; COUNT = 10
Congratulations on successfully guiding the missionaries and cannibals across the river!
Solution path:

M on left:3
C on left:3
M on right:0
C on right:0
boat is on the left.

M on left:2
C on left:2
M on right:1
C on right:1
boat is on the right.

M on left:3
C on left:2
M on right:0
C on right:1
boat is on the left.

M on left:1
C on left:1
M on right:2
C on right:2
boat is on the right.

M on left:3
C on left:1
M on right:0
C on right:2

boat is on the left.

M on left:0
C on left:1
  M on right:3
  C on right:2
boat is on the right.

M on left:1
C on left:1
  M on right:2
  C on right:2
boat is on the left.

M on left:0
C on left:0
  M on right:3
  C on right:3
boat is on the right.

Length of solution path found: 7 edges
10 states expanded.
MAX_OPEN_LENGTH = 2

# Farmer_Fox with ItrDFS

Welcome to ItrDFS
Initial State:

 On the left: fox chicken grain
 On the right:
 farmer and boat are on the left.

len(OPEN)=1; len(CLOSED)=0; COUNT = 0
OPEN is now:
 On the left: fox grain
 On the right: chicken
 farmer and boat are on the right.

len(OPEN)=1; len(CLOSED)=1; COUNT = 1
OPEN is now:
 On the left: fox grain
 On the right: chicken
 farmer and boat are on the left.

len(OPEN)=1; len(CLOSED)=2; COUNT = 2
OPEN is now:
 On the left: grain
 On the right: fox chicken
 farmer and boat are on the right.
,
 On the left: fox
 On the right: chicken grain
 farmer and boat are on the right.

len(OPEN)=2; len(CLOSED)=3; COUNT = 3
OPEN is now:
 On the left: chicken grain
 On the right: fox
 farmer and boat are on the left.
,
 On the left: fox
 On the right: chicken grain
 farmer and boat are on the right.

len(OPEN)=2; len(CLOSED)=4; COUNT = 4
OPEN is now:
 On the left: chicken
 On the right: fox grain
 farmer and boat are on the right.
,
 On the left: fox
 On the right: chicken grain
 farmer and boat are on the right.

len(OPEN)=2; len(CLOSED)=5; COUNT = 5

OPEN is now:
 On the left: fox chicken
 On the right: grain
 farmer and boat are on the left.
,
 On the left: chicken
 On the right: fox grain
 farmer and boat are on the left.
,
 On the left: fox
 On the right: chicken grain
 farmer and boat are on the right.

len(OPEN)=3; len(CLOSED)=6; COUNT = 6
OPEN is now:
 On the left: fox
 On the right: chicken grain
 farmer and boat are on the right.
,
 On the left: chicken
 On the right: fox grain
 farmer and boat are on the left.

len(OPEN)=2; len(CLOSED)=7; COUNT = 7
OPEN is now:
 On the left: chicken
 On the right: fox grain
 farmer and boat are on the left.

len(OPEN)=1; len(CLOSED)=8; COUNT = 8
OPEN is now:
 On the left:
 On the right: fox chicken grain
 farmer and boat are on the right.

len(OPEN)=1; len(CLOSED)=9; COUNT = 9
Congratulations on successfully guiding the fox chicken and grain across the river!
Solution path:

 On the left: fox chicken grain
 On the right:
 farmer and boat are on the left.


 On the left: fox grain
 On the right: chicken
 farmer and boat are on the right.


 On the left: fox grain
 On the right: chicken
 farmer and boat are on the left.

On the left: grain
On the right: fox chicken
farmer and boat are on the right.

On the left: chicken grain
On the right: fox
farmer and boat are on the left.

On the left: chicken
On the right: fox grain
farmer and boat are on the right.

On the left: chicken
On the right: fox grain
farmer and boat are on the left.

On the left:
On the right: fox chicken grain
farmer and boat are on the right.

Length of solution path found: 7 edges
9 states expanded.
MAX_OPEN_LENGTH = 3

# Farmer_Fox with BFS

Welcome to BFS
Initial State:

 On the left: fox chicken grain
 On the right:
 farmer and boat are on the left.

len(OPEN)=1; len(CLOSED)=0; COUNT = 0
OPEN is now:
 On the left: fox grain
 On the right: chicken
 farmer and boat are on the right.

len(OPEN)=1; len(CLOSED)=1; COUNT = 1
OPEN is now:
 On the left: fox grain
 On the right: chicken
 farmer and boat are on the left.

len(OPEN)=1; len(CLOSED)=2; COUNT = 2
OPEN is now:
 On the left: grain
 On the right: fox chicken
 farmer and boat are on the right.
,
 On the left: fox
 On the right: chicken grain
 farmer and boat are on the right.

len(OPEN)=2; len(CLOSED)=3; COUNT = 3
OPEN is now:
 On the left: fox
 On the right: chicken grain
 farmer and boat are on the right.
,
 On the left: chicken grain
 On the right: fox
 farmer and boat are on the left.

len(OPEN)=2; len(CLOSED)=4; COUNT = 4
OPEN is now:
 On the left: chicken grain
 On the right: fox
 farmer and boat are on the left.
,
 On the left: fox chicken
 On the right: grain
 farmer and boat are on the left.

len(OPEN)=2; len(CLOSED)=5; COUNT = 5

OPEN is now:
 On the left: fox chicken
 On the right: grain
 farmer and boat are on the left.
,
 On the left: chicken
 On the right: fox grain
 farmer and boat are on the right.

len(OPEN)=2; len(CLOSED)=6; COUNT = 6
OPEN is now:
 On the left: chicken
 On the right: fox grain
 farmer and boat are on the right.

len(OPEN)=1; len(CLOSED)=7; COUNT = 7
OPEN is now:
 On the left: chicken
 On the right: fox grain
 farmer and boat are on the left.

len(OPEN)=1; len(CLOSED)=8; COUNT = 8
OPEN is now:
 On the left:
 On the right: fox chicken grain
 farmer and boat are on the right.

len(OPEN)=1; len(CLOSED)=9; COUNT = 9
Congratulations on successfully guiding the fox chicken and grain across the river!
Solution path:

 On the left: fox chicken grain
 On the right:
 farmer and boat are on the left.


 On the left: fox grain
 On the right: chicken
 farmer and boat are on the right.


 On the left: fox grain
 On the right: chicken
 farmer and boat are on the left.


 On the left: fox
 On the right: chicken grain
 farmer and boat are on the right.


 On the left: fox chicken
 On the right: grain
 farmer and boat are on the left.

On the left: chicken
On the right: fox grain
farmer and boat are on the right.

On the left: chicken
On the right: fox grain
farmer and boat are on the left.

On the left:
On the right: fox chicken grain
farmer and boat are on the right.

Length of solution path found: 7 edges
9 states expanded.
MAX_OPEN_LENGTH = 2

# HanoiTower with ItrDFS

Welcome to ItrDFS
Initial State:
[[4, 3, 2, 1] ,[] ,[]]
len(OPEN)=1; len(CLOSED)=0; COUNT = 0
OPEN is now: [[4, 3, 2] ,[1] ,[]], [[4, 3, 2] ,[] ,[1]]
len(OPEN)=2; len(CLOSED)=1; COUNT = 1
OPEN is now: [[4, 3] ,[1] ,[2]], [[4, 3, 2] ,[] ,[1]]
len(OPEN)=2; len(CLOSED)=2; COUNT = 2
OPEN is now: [[4, 3, 1] ,[] ,[2]], [[4, 3] ,[] ,[2, 1]], [[4, 3, 2] ,[] ,[1]]
len(OPEN)=3; len(CLOSED)=3; COUNT = 3
OPEN is now: [[4, 3] ,[] ,[2, 1]], [[4, 3, 1] ,[2] ,[]], [[4, 3, 2] ,[] ,[1]]
len(OPEN)=3; len(CLOSED)=4; COUNT = 4
OPEN is now: [[4] ,[3] ,[2, 1]], [[4, 3, 1] ,[2] ,[]], [[4, 3, 2] ,[] ,[1]]
len(OPEN)=3; len(CLOSED)=5; COUNT = 5
OPEN is now: [[4, 1] ,[3] ,[2]], [[4] ,[3, 1] ,[2]], [[4, 3, 1] ,[2] ,[]], [[4, 3, 2] ,[] ,[1]]
len(OPEN)=4; len(CLOSED)=6; COUNT = 6
OPEN is now: [[4] ,[3, 1] ,[2]], [[4, 1] ,[3, 2] ,[]], [[4, 3, 1] ,[2] ,[]], [[4, 3, 2] ,[] ,[1]]
len(OPEN)=4; len(CLOSED)=7; COUNT = 7
OPEN is now: [[4, 2] ,[3, 1] ,[]], [[4, 1] ,[3, 2] ,[]], [[4, 3, 1] ,[2] ,[]], [[4, 3, 2] ,[] ,[1]]
len(OPEN)=4; len(CLOSED)=8; COUNT = 8
OPEN is now: [[4, 2, 1] ,[3] ,[]], [[4, 2] ,[3] ,[1]], [[4, 1] ,[3, 2] ,[]], [[4, 3, 1] ,[2] ,[]], [[4, 3, 2] ,[] ,[1]]
len(OPEN)=5; len(CLOSED)=9; COUNT = 9
OPEN is now: [[4, 2] ,[3] ,[1]], [[4, 2, 1] ,[] ,[3]], [[4, 1] ,[3, 2] ,[]], [[4, 3, 1] ,[2] ,[]], [[4, 3, 2] ,[] ,[1]]
len(OPEN)=5; len(CLOSED)=10; COUNT = 10
OPEN is now: [[4] ,[3, 2] ,[1]], [[4, 2, 1] ,[] ,[3]], [[4, 1] ,[3, 2] ,[]], [[4, 3, 1] ,[2] ,[]], [[4, 3, 2] ,[] ,[1]]
len(OPEN)=5; len(CLOSED)=11; COUNT = 11
OPEN is now: [[4, 1] ,[3, 2] ,[]], [[4] ,[3, 2, 1] ,[]], [[4, 2, 1] ,[] ,[3]], [[4, 3, 1] ,[2] ,[]], [[4, 3, 2] ,[] ,[1]]
len(OPEN)=5; len(CLOSED)=12; COUNT = 12
OPEN is now: [[4] ,[3, 2, 1] ,[]], [[4, 2, 1] ,[] ,[3]], [[4, 3, 1] ,[2] ,[]], [[4, 3, 2] ,[] ,[1]]
len(OPEN)=4; len(CLOSED)=13; COUNT = 13
OPEN is now: [[] ,[3, 2, 1] ,[4]], [[4, 2, 1] ,[] ,[3]], [[4, 3, 1] ,[2] ,[]], [[4, 3, 2] ,[] ,[1]]
len(OPEN)=4; len(CLOSED)=14; COUNT = 14
OPEN is now: [[1] ,[3, 2] ,[4]], [[] ,[3, 2] ,[4, 1]], [[4, 2, 1] ,[] ,[3]], [[4, 3, 1] ,[2] ,[]], [[4, 3, 2] ,[] ,[1]]
len(OPEN)=5; len(CLOSED)=15; COUNT = 15
OPEN is now: [[] ,[3, 2] ,[4, 1]], [[1] ,[3] ,[4, 2]], [[4, 2, 1] ,[] ,[3]], [[4, 3, 1] ,[2] ,[]], [[4, 3, 2] ,[] ,[1]]
len(OPEN)=5; len(CLOSED)=16; COUNT = 16
OPEN is now: [[2] ,[3] ,[4, 1]], [[1] ,[3] ,[4, 2]], [[4, 2, 1] ,[] ,[3]], [[4, 3, 1] ,[2] ,[]], [[4, 3, 2] ,[] ,[1]]
len(OPEN)=5; len(CLOSED)=17; COUNT = 17
OPEN is now: [[2, 1] ,[3] ,[4]], [[2] ,[3, 1] ,[4]], [[1] ,[3] ,[4, 2]], [[4, 2, 1] ,[] ,[3]], [[4, 3, 1] ,[2] ,[]], [[4, 3, 2] ,[] ,[1]]
len(OPEN)=6; len(CLOSED)=18; COUNT = 18
OPEN is now: [[2] ,[3, 1] ,[4]], [[2, 1] ,[] ,[4, 3]], [[1] ,[3] ,[4, 2]], [[4, 2, 1] ,[] ,[3]], [[4, 3, 1] ,[2] ,[]], [[4, 3, 2] ,[] ,[1]]
len(OPEN)=6; len(CLOSED)=19; COUNT = 19
OPEN is now: [[] ,[3, 1] ,[4, 2]], [[2, 1] ,[] ,[4, 3]], [[1] ,[3] ,[4, 2]], [[4, 2, 1] ,[] ,[3]], [[4, 3, 1] ,[2] ,[]], [[4, 3, 2] ,[] ,[1]]
len(OPEN)=6; len(CLOSED)=20; COUNT = 20
OPEN is now: [[1] ,[3] ,[4, 2]], [[] ,[3] ,[4, 2, 1]], [[2, 1] ,[] ,[4, 3]], [[4, 2, 1] ,[] ,[3]], [[4, 3, 1] ,[2] ,[]], [[4, 3, 2] ,[] ,[1]]

len(OPEN)=6; len(CLOSED)=21; COUNT = 21
OPEN is now: [[] ,[3] ,[4, 2, 1]], [[2, 1] ,[] ,[4, 3]], [[4, 2, 1] ,[] ,[3]], [[4, 3, 1] ,[2] ,[]], [[4, 3, 2] ,[] ,[1]]
len(OPEN)=5; len(CLOSED)=22; COUNT = 22
OPEN is now: [[3] ,[] ,[4, 2, 1]], [[2, 1] ,[] ,[4, 3]], [[4, 2, 1] ,[] ,[3]], [[4, 3, 1] ,[2] ,[]], [[4, 3, 2] ,[] ,[1]]
len(OPEN)=5; len(CLOSED)=23; COUNT = 23
OPEN is now: [[3, 1] ,[] ,[4, 2]], [[3] ,[1] ,[4, 2]], [[2, 1] ,[] ,[4, 3]], [[4, 2, 1] ,[] ,[3]], [[4, 3, 1] ,[2] ,[]], [[4, 3, 2] ,[] ,[1]]
len(OPEN)=6; len(CLOSED)=24; COUNT = 24
OPEN is now: [[3] ,[1] ,[4, 2]], [[3, 1] ,[2] ,[4]], [[2, 1] ,[] ,[4, 3]], [[4, 2, 1] ,[] ,[3]], [[4, 3, 1] ,[2] ,[]], [[4, 3, 2] ,[] ,[1]]
len(OPEN)=6; len(CLOSED)=25; COUNT = 25
OPEN is now: [[3, 2] ,[1] ,[4]], [[3, 1] ,[2] ,[4]], [[2, 1] ,[] ,[4, 3]], [[4, 2, 1] ,[] ,[3]], [[4, 3, 1] ,[2] ,[]], [[4, 3, 2] ,[] ,[1]]
len(OPEN)=6; len(CLOSED)=26; COUNT = 26
OPEN is now: [[3, 2, 1] ,[] ,[4]], [[3, 2] ,[] ,[4, 1]], [[3, 1] ,[2] ,[4]], [[2, 1] ,[] ,[4, 3]], [[4, 2, 1] ,[] ,[3]], [[4, 3, 1] ,[2] ,[]], [[4, 3, 2] ,[] ,[1]]
len(OPEN)=7; len(CLOSED)=27; COUNT = 27
OPEN is now: [[3, 2] ,[] ,[4, 1]], [[3, 2, 1] ,[4] ,[]], [[3, 1] ,[2] ,[4]], [[2, 1] ,[] ,[4, 3]], [[4, 2, 1] ,[] ,[3]], [[4, 3, 1] ,[2] ,[]], [[4, 3, 2] ,[] ,[1]]
len(OPEN)=7; len(CLOSED)=28; COUNT = 28
OPEN is now: [[3] ,[2] ,[4, 1]], [[3, 2, 1] ,[4] ,[]], [[3, 1] ,[2] ,[4]], [[2, 1] ,[] ,[4, 3]], [[4, 2, 1] ,[] ,[3]], [[4, 3, 1] ,[2] ,[]], [[4, 3, 2] ,[] ,[1]]
len(OPEN)=7; len(CLOSED)=29; COUNT = 29
OPEN is now: [[3, 1] ,[2] ,[4]], [[3] ,[2, 1] ,[4]], [[3, 2, 1] ,[4] ,[]], [[2, 1] ,[] ,[4, 3]], [[4, 2, 1] ,[] ,[3]], [[4, 3, 1] ,[2] ,[]], [[4, 3, 2] ,[] ,[1]]
len(OPEN)=7; len(CLOSED)=30; COUNT = 30
OPEN is now: [[3] ,[2, 1] ,[4]], [[3, 2, 1] ,[4] ,[]], [[2, 1] ,[] ,[4, 3]], [[4, 2, 1] ,[] ,[3]], [[4, 3, 1] ,[2] ,[]], [[4, 3, 2] ,[] ,[1]]
len(OPEN)=6; len(CLOSED)=31; COUNT = 31
OPEN is now: [[] ,[2, 1] ,[4, 3]], [[3, 2, 1] ,[4] ,[]], [[2, 1] ,[] ,[4, 3]], [[4, 2, 1] ,[] ,[3]], [[4, 3, 1] ,[2] ,[]], [[4, 3, 2] ,[] ,[1]]
len(OPEN)=6; len(CLOSED)=32; COUNT = 32
OPEN is now: [[1] ,[2] ,[4, 3]], [[] ,[2] ,[4, 3, 1]], [[3, 2, 1] ,[4] ,[]], [[2, 1] ,[] ,[4, 3]], [[4, 2, 1] ,[] ,[3]], [[4, 3, 1] ,[2] ,[]], [[4, 3, 2] ,[] ,[1]]
len(OPEN)=7; len(CLOSED)=33; COUNT = 33
OPEN is now: [[] ,[2] ,[4, 3, 1]], [[1] ,[] ,[4, 3, 2]], [[3, 2, 1] ,[4] ,[]], [[2, 1] ,[] ,[4, 3]], [[4, 2, 1] ,[] ,[3]], [[4, 3, 1] ,[2] ,[]], [[4, 3, 2] ,[] ,[1]]
len(OPEN)=7; len(CLOSED)=34; COUNT = 34
OPEN is now: [[2] ,[] ,[4, 3, 1]], [[1] ,[] ,[4, 3, 2]], [[3, 2, 1] ,[4] ,[]], [[2, 1] ,[] ,[4, 3]], [[4, 2, 1] ,[] ,[3]], [[4, 3, 1] ,[2] ,[]], [[4, 3, 2] ,[] ,[1]]
len(OPEN)=7; len(CLOSED)=35; COUNT = 35
OPEN is now: [[2, 1] ,[] ,[4, 3]], [[2] ,[1] ,[4, 3]], [[1] ,[] ,[4, 3, 2]], [[3, 2, 1] ,[4] ,[]], [[4, 2, 1] ,[] ,[3]], [[4, 3, 1] ,[2] ,[]], [[4, 3, 2] ,[] ,[1]]
len(OPEN)=7; len(CLOSED)=36; COUNT = 36
OPEN is now: [[2] ,[1] ,[4, 3]], [[1] ,[] ,[4, 3, 2]], [[3, 2, 1] ,[4] ,[]], [[4, 2, 1] ,[] ,[3]], [[4, 3, 1] ,[2] ,[]], [[4, 3, 2] ,[] ,[1]]
len(OPEN)=6; len(CLOSED)=37; COUNT = 37
OPEN is now: [[] ,[1] ,[4, 3, 2]], [[1] ,[] ,[4, 3, 2]], [[3, 2, 1] ,[4] ,[]], [[4, 2, 1] ,[] ,[3]], [[4, 3, 1] ,[2] ,[]], [[4, 3, 2] ,[] ,[1]]
len(OPEN)=6; len(CLOSED)=38; COUNT = 38
OPEN is now: [[1] ,[] ,[4, 3, 2]], [[] ,[] ,[4, 3, 2, 1]], [[3, 2, 1] ,[4] ,[]], [[4, 2, 1] ,[] ,[3]], [[4, 3, 1] ,[2] ,[]], [[4, 3, 2] ,[] ,[1]]
len(OPEN)=6; len(CLOSED)=39; COUNT = 39
OPEN is now: [[] ,[] ,[4, 3, 2, 1]], [[3, 2, 1] ,[4] ,[]], [[4, 2, 1] ,[] ,[3]], [[4, 3, 1] ,[2] ,[]], [[4, 3, 2] ,[] ,[1]]

len(OPEN)=5; len(CLOSED)=40; COUNT = 40
The Tower Transport is Triumphant!
Solution path:
[[4, 3, 2, 1] ,[] ,[]]
[[4, 3, 2] ,[1] ,[]]
[[4, 3] ,[1] ,[2]]
[[4, 3, 1] ,[] ,[2]]
[[4, 3] ,[] ,[2, 1]]
[[4] ,[3] ,[2, 1]]
[[4, 1] ,[3] ,[2]]
[[4] ,[3, 1] ,[2]]
[[4, 2] ,[3, 1] ,[]]
[[4, 2, 1] ,[3] ,[]]
[[4, 2] ,[3] ,[1]]
[[4] ,[3, 2] ,[1]]
[[4, 1] ,[3, 2] ,[]]
[[4] ,[3, 2, 1] ,[]]
[[] ,[3, 2, 1] ,[4]]
[[1] ,[3, 2] ,[4]]
[[] ,[3, 2] ,[4, 1]]
[[2] ,[3] ,[4, 1]]
[[2, 1] ,[3] ,[4]]
[[2] ,[3, 1] ,[4]]
[[] ,[3, 1] ,[4, 2]]
[[1] ,[3] ,[4, 2]]
[[] ,[3] ,[4, 2, 1]]
[[3] ,[] ,[4, 2, 1]]
[[3, 1] ,[] ,[4, 2]]
[[3] ,[1] ,[4, 2]]
[[3, 2] ,[1] ,[4]]
[[3, 2, 1] ,[] ,[4]]
[[3, 2] ,[] ,[4, 1]]
[[3] ,[2] ,[4, 1]]
[[3, 1] ,[2] ,[4]]
[[3] ,[2, 1] ,[4]]
[[] ,[2, 1] ,[4, 3]]
[[1] ,[2] ,[4, 3]]
[[] ,[2] ,[4, 3, 1]]
[[2] ,[] ,[4, 3, 1]]
[[2, 1] ,[] ,[4, 3]]
[[2] ,[1] ,[4, 3]]
[[] ,[1] ,[4, 3, 2]]
[[1] ,[] ,[4, 3, 2]]
[[] ,[] ,[4, 3, 2, 1]]
Length of solution path found: 40 edges
40 states expanded.
MAX_OPEN_LENGTH = 7

# HanoiTower with BFS

Welcome to BFS
Initial State:
[[4, 3, 2, 1] ,[] ,[]]
len(OPEN)=1; len(CLOSED)=0; COUNT = 0
OPEN is now: [[4, 3, 2] ,[1] ,[]], [[4, 3, 2] ,[] ,[1]]
len(OPEN)=2; len(CLOSED)=1; COUNT = 1
OPEN is now: [[4, 3, 2] ,[] ,[1]], [[4, 3] ,[1] ,[2]]
len(OPEN)=2; len(CLOSED)=2; COUNT = 2
OPEN is now: [[4, 3] ,[1] ,[2]], [[4, 3] ,[2] ,[1]]
len(OPEN)=2; len(CLOSED)=3; COUNT = 3
OPEN is now: [[4, 3] ,[2] ,[1]], [[4, 3, 1] ,[] ,[2]], [[4, 3] ,[] ,[2, 1]]
len(OPEN)=3; len(CLOSED)=4; COUNT = 4
OPEN is now: [[4, 3, 1] ,[] ,[2]], [[4, 3] ,[] ,[2, 1]], [[4, 3, 1] ,[2] ,[]], [[4, 3] ,[2, 1] ,[]]
len(OPEN)=4; len(CLOSED)=5; COUNT = 5
OPEN is now: [[4, 3] ,[] ,[2, 1]], [[4, 3, 1] ,[2] ,[]], [[4, 3] ,[2, 1] ,[]]
len(OPEN)=3; len(CLOSED)=6; COUNT = 6
OPEN is now: [[4, 3, 1] ,[2] ,[]], [[4, 3] ,[2, 1] ,[]], [[4] ,[3] ,[2, 1]]
len(OPEN)=3; len(CLOSED)=7; COUNT = 7
OPEN is now: [[4, 3] ,[2, 1] ,[]], [[4] ,[3] ,[2, 1]]
len(OPEN)=2; len(CLOSED)=8; COUNT = 8
OPEN is now: [[4] ,[3] ,[2, 1]], [[4] ,[2, 1] ,[3]]
len(OPEN)=2; len(CLOSED)=9; COUNT = 9
OPEN is now: [[4] ,[2, 1] ,[3]], [[4, 1] ,[3] ,[2]], [[4] ,[3, 1] ,[2]]
len(OPEN)=3; len(CLOSED)=10; COUNT = 10
OPEN is now: [[4, 1] ,[3] ,[2]], [[4] ,[3, 1] ,[2]], [[4, 1] ,[2] ,[3]], [[4] ,[2] ,[3, 1]]
len(OPEN)=4; len(CLOSED)=11; COUNT = 11
OPEN is now: [[4] ,[3, 1] ,[2]], [[4, 1] ,[2] ,[3]], [[4] ,[2] ,[3, 1]], [[4, 1] ,[3, 2] ,[]]
len(OPEN)=4; len(CLOSED)=12; COUNT = 12
OPEN is now: [[4, 1] ,[2] ,[3]], [[4] ,[2] ,[3, 1]], [[4, 1] ,[3, 2] ,[]], [[4, 2] ,[3, 1] ,[]]
len(OPEN)=4; len(CLOSED)=13; COUNT = 13
OPEN is now: [[4] ,[2] ,[3, 1]], [[4, 1] ,[3, 2] ,[]], [[4, 2] ,[3, 1] ,[]], [[4, 1] ,[] ,[3, 2]]
len(OPEN)=4; len(CLOSED)=14; COUNT = 14
OPEN is now: [[4, 1] ,[3, 2] ,[]], [[4, 2] ,[3, 1] ,[]], [[4, 1] ,[] ,[3, 2]], [[4, 2] ,[] ,[3, 1]]
len(OPEN)=4; len(CLOSED)=15; COUNT = 15
OPEN is now: [[4, 2] ,[3, 1] ,[]], [[4, 1] ,[] ,[3, 2]], [[4, 2] ,[] ,[3, 1]], [[4] ,[3, 2, 1] ,[]], [[4] ,[3, 2] ,[1]]
len(OPEN)=5; len(CLOSED)=16; COUNT = 16
OPEN is now: [[4, 1] ,[] ,[3, 2]], [[4, 2] ,[] ,[3, 1]], [[4] ,[3, 2, 1] ,[]], [[4] ,[3, 2] ,[1]], [[4, 2, 1] ,[3] ,[]],
[[4, 2] ,[3] ,[1]]
len(OPEN)=6; len(CLOSED)=17; COUNT = 17
OPEN is now: [[4, 2] ,[] ,[3, 1]], [[4] ,[3, 2, 1] ,[]], [[4] ,[3, 2] ,[1]], [[4, 2, 1] ,[3] ,[]], [[4, 2] ,[3] ,[1]],
[[4] ,[1] ,[3, 2]], [[4] ,[] ,[3, 2, 1]]
len(OPEN)=7; len(CLOSED)=18; COUNT = 18
OPEN is now: [[4] ,[3, 2, 1] ,[]], [[4] ,[3, 2] ,[1]], [[4, 2, 1] ,[3] ,[]], [[4, 2] ,[3] ,[1]], [[4] ,[1] ,[3, 2]], [[4] ,
[] ,[3, 2, 1]], [[4, 2, 1] ,[] ,[3]], [[4, 2] ,[1] ,[3]]
len(OPEN)=8; len(CLOSED)=19; COUNT = 19
OPEN is now: [[4] ,[3, 2] ,[1]], [[4, 2, 1] ,[3] ,[]], [[4, 2] ,[3] ,[1]], [[4] ,[1] ,[3, 2]], [[4] ,[] ,[3, 2, 1]], [[4,
2, 1] ,[] ,[3]], [[4, 2] ,[1] ,[3]], [[] ,[3, 2, 1] ,[4]]
len(OPEN)=8; len(CLOSED)=20; COUNT = 20

OPEN is now: [[4, 2, 1] ,[3] ,[]], [[4, 2] ,[3] ,[1]], [[4] ,[1] ,[3, 2]], [[4] ,[] ,[3, 2, 1]], [[4, 2, 1] ,[] ,[3]], [[4, 2] ,[1] ,[3]], [[] ,[3, 2, 1] ,[4]]
len(OPEN)=7; len(CLOSED)=21; COUNT = 21
OPEN is now: [[4, 2] ,[3] ,[1]], [[4] ,[1] ,[3, 2]], [[4] ,[] ,[3, 2, 1]], [[4, 2, 1] ,[] ,[3]], [[4, 2] ,[1] ,[3]], [[] ,[3, 2, 1] ,[4]]
len(OPEN)=6; len(CLOSED)=22; COUNT = 22
OPEN is now: [[4] ,[1] ,[3, 2]], [[4] ,[] ,[3, 2, 1]], [[4, 2, 1] ,[] ,[3]], [[4, 2] ,[1] ,[3]], [[] ,[3, 2, 1] ,[4]]
len(OPEN)=5; len(CLOSED)=23; COUNT = 23
OPEN is now: [[4] ,[] ,[3, 2, 1]], [[4, 2, 1] ,[] ,[3]], [[4, 2] ,[1] ,[3]], [[] ,[3, 2, 1] ,[4]]
len(OPEN)=4; len(CLOSED)=24; COUNT = 24
OPEN is now: [[4, 2, 1] ,[] ,[3]], [[4, 2] ,[1] ,[3]], [[] ,[3, 2, 1] ,[4]], [[] ,[4] ,[3, 2, 1]]
len(OPEN)=4; len(CLOSED)=25; COUNT = 25
OPEN is now: [[4, 2] ,[1] ,[3]], [[] ,[3, 2, 1] ,[4]], [[] ,[4] ,[3, 2, 1]]
len(OPEN)=3; len(CLOSED)=26; COUNT = 26
OPEN is now: [[] ,[3, 2, 1] ,[4]], [[] ,[4] ,[3, 2, 1]]
len(OPEN)=2; len(CLOSED)=27; COUNT = 27
OPEN is now: [[] ,[4] ,[3, 2, 1]], [[1] ,[3, 2] ,[4]], [[] ,[3, 2] ,[4, 1]]
len(OPEN)=3; len(CLOSED)=28; COUNT = 28
OPEN is now: [[1] ,[3, 2] ,[4]], [[] ,[3, 2] ,[4, 1]], [[1] ,[4] ,[3, 2]], [[] ,[4, 1] ,[3, 2]]
len(OPEN)=4; len(CLOSED)=29; COUNT = 29
OPEN is now: [[] ,[3, 2] ,[4, 1]], [[1] ,[4] ,[3, 2]], [[] ,[4, 1] ,[3, 2]], [[1] ,[3] ,[4, 2]]
len(OPEN)=4; len(CLOSED)=30; COUNT = 30
OPEN is now: [[1] ,[4] ,[3, 2]], [[] ,[4, 1] ,[3, 2]], [[1] ,[3] ,[4, 2]], [[2] ,[3] ,[4, 1]]
len(OPEN)=4; len(CLOSED)=31; COUNT = 31
OPEN is now: [[] ,[4, 1] ,[3, 2]], [[1] ,[3] ,[4, 2]], [[2] ,[3] ,[4, 1]], [[1] ,[4, 2] ,[3]]
len(OPEN)=4; len(CLOSED)=32; COUNT = 32
OPEN is now: [[1] ,[3] ,[4, 2]], [[2] ,[3] ,[4, 1]], [[1] ,[4, 2] ,[3]], [[2] ,[4, 1] ,[3]]
len(OPEN)=4; len(CLOSED)=33; COUNT = 33
OPEN is now: [[2] ,[3] ,[4, 1]], [[1] ,[4, 2] ,[3]], [[2] ,[4, 1] ,[3]], [[] ,[3, 1] ,[4, 2]], [[] ,[3] ,[4, 2, 1]]
len(OPEN)=5; len(CLOSED)=34; COUNT = 34
OPEN is now: [[1] ,[4, 2] ,[3]], [[2] ,[4, 1] ,[3]], [[] ,[3, 1] ,[4, 2]], [[] ,[3] ,[4, 2, 1]], [[2, 1] ,[3] ,[4]], [[2] ,[3, 1] ,[4]]
len(OPEN)=6; len(CLOSED)=35; COUNT = 35
OPEN is now: [[2] ,[4, 1] ,[3]], [[] ,[3, 1] ,[4, 2]], [[] ,[3] ,[4, 2, 1]], [[2, 1] ,[3] ,[4]], [[2] ,[3, 1] ,[4]], [[] ,[4, 2, 1] ,[3]], [[] ,[4, 2] ,[3, 1]]
len(OPEN)=7; len(CLOSED)=36; COUNT = 36
OPEN is now: [[] ,[3, 1] ,[4, 2]], [[] ,[3] ,[4, 2, 1]], [[2, 1] ,[3] ,[4]], [[2] ,[3, 1] ,[4]], [[] ,[4, 2, 1] ,[3]], [[] ,[4, 2] ,[3, 1]], [[2, 1] ,[4] ,[3]], [[2] ,[4] ,[3, 1]]
len(OPEN)=8; len(CLOSED)=37; COUNT = 37
OPEN is now: [[] ,[3] ,[4, 2, 1]], [[2, 1] ,[3] ,[4]], [[2] ,[3, 1] ,[4]], [[] ,[4, 2, 1] ,[3]], [[] ,[4, 2] ,[3, 1]], [[2, 1] ,[4] ,[3]], [[2] ,[4] ,[3, 1]]
len(OPEN)=7; len(CLOSED)=38; COUNT = 38
OPEN is now: [[2, 1] ,[3] ,[4]], [[2] ,[3, 1] ,[4]], [[] ,[4, 2, 1] ,[3]], [[] ,[4, 2] ,[3, 1]], [[2, 1] ,[4] ,[3]], [[2] ,[4] ,[3, 1]], [[3] ,[] ,[4, 2, 1]]
len(OPEN)=7; len(CLOSED)=39; COUNT = 39
OPEN is now: [[2] ,[3, 1] ,[4]], [[] ,[4, 2, 1] ,[3]], [[] ,[4, 2] ,[3, 1]], [[2, 1] ,[4] ,[3]], [[2] ,[4] ,[3, 1]], [[3] ,[] ,[4, 2, 1]], [[2, 1] ,[] ,[4, 3]]
len(OPEN)=7; len(CLOSED)=40; COUNT = 40
OPEN is now: [[] ,[4, 2, 1] ,[3]], [[] ,[4, 2] ,[3, 1]], [[2, 1] ,[4] ,[3]], [[2] ,[4] ,[3, 1]], [[3] ,[] ,[4, 2, 1]], [[2, 1] ,[] ,[4, 3]]
len(OPEN)=6; len(CLOSED)=41; COUNT = 41
OPEN is now: [[] ,[4, 2] ,[3, 1]], [[2, 1] ,[4] ,[3]], [[2] ,[4] ,[3, 1]], [[3] ,[] ,[4, 2, 1]], [[2, 1] ,[] ,[4, 3]], [[3] ,[4, 2, 1] ,[]]
len(OPEN)=6; len(CLOSED)=42; COUNT = 42

OPEN is now: [[2, 1] ,[4] ,[3]], [[2] ,[4] ,[3, 1]], [[3] ,[] ,[4, 2, 1]], [[2, 1] ,[] ,[4, 3]], [[3] ,[4, 2, 1] ,[]]
len(OPEN)=5; len(CLOSED)=43; COUNT = 43
OPEN is now: [[2] ,[4] ,[3, 1]], [[3] ,[] ,[4, 2, 1]], [[2, 1] ,[] ,[4, 3]], [[3] ,[4, 2, 1] ,[]], [[2, 1] ,[4, 3] ,[]]
len(OPEN)=5; len(CLOSED)=44; COUNT = 44
OPEN is now: [[3] ,[] ,[4, 2, 1]], [[2, 1] ,[] ,[4, 3]], [[3] ,[4, 2, 1] ,[]], [[2, 1] ,[4, 3] ,[]]
len(OPEN)=4; len(CLOSED)=45; COUNT = 45
OPEN is now: [[2, 1] ,[] ,[4, 3]], [[3] ,[4, 2, 1] ,[]], [[2, 1] ,[4, 3] ,[]], [[3, 1] ,[] ,[4, 2]], [[3] ,[1] ,[4, 2]]
len(OPEN)=5; len(CLOSED)=46; COUNT = 46
OPEN is now: [[3] ,[4, 2, 1] ,[]], [[2, 1] ,[4, 3] ,[]], [[3, 1] ,[] ,[4, 2]], [[3] ,[1] ,[4, 2]], [[2] ,[1] ,[4, 3]],
[[2] ,[] ,[4, 3, 1]]
len(OPEN)=6; len(CLOSED)=47; COUNT = 47
OPEN is now: [[2, 1] ,[4, 3] ,[]], [[3, 1] ,[] ,[4, 2]], [[3] ,[1] ,[4, 2]], [[2] ,[1] ,[4, 3]], [[2] ,[] ,[4, 3, 1]], [[3,
1] ,[4, 2] ,[]], [[3] ,[4, 2] ,[1]]
len(OPEN)=7; len(CLOSED)=48; COUNT = 48
OPEN is now: [[3, 1] ,[] ,[4, 2]], [[3] ,[1] ,[4, 2]], [[2] ,[1] ,[4, 3]], [[2] ,[] ,[4, 3, 1]], [[3, 1] ,[4, 2] ,[]],
[[3] ,[4, 2] ,[1]], [[2] ,[4, 3, 1] ,[]], [[2] ,[4, 3] ,[1]]
len(OPEN)=8; len(CLOSED)=49; COUNT = 49
OPEN is now: [[3] ,[1] ,[4, 2]], [[2] ,[1] ,[4, 3]], [[2] ,[] ,[4, 3, 1]], [[3, 1] ,[4, 2] ,[]], [[3] ,[4, 2] ,[1]], [[2] ,
[4, 3, 1] ,[]], [[2] ,[4, 3] ,[1]], [[3, 1] ,[2] ,[4]]
len(OPEN)=8; len(CLOSED)=50; COUNT = 50
OPEN is now: [[2] ,[1] ,[4, 3]], [[2] ,[] ,[4, 3, 1]], [[3, 1] ,[4, 2] ,[]], [[3] ,[4, 2] ,[1]], [[2] ,[4, 3, 1] ,[]],
[[2] ,[4, 3] ,[1]], [[3, 1] ,[2] ,[4]], [[3, 2] ,[1] ,[4]]
len(OPEN)=8; len(CLOSED)=51; COUNT = 51
OPEN is now: [[2] ,[] ,[4, 3, 1]], [[3, 1] ,[4, 2] ,[]], [[3] ,[4, 2] ,[1]], [[2] ,[4, 3, 1] ,[]], [[2] ,[4, 3] ,[1]], [[3,
1] ,[2] ,[4]], [[3, 2] ,[1] ,[4]], [[] ,[1] ,[4, 3, 2]]
len(OPEN)=8; len(CLOSED)=52; COUNT = 52
OPEN is now: [[3, 1] ,[4, 2] ,[]], [[3] ,[4, 2] ,[1]], [[2] ,[4, 3, 1] ,[]], [[2] ,[4, 3] ,[1]], [[3, 1] ,[2] ,[4]], [[3,
2] ,[1] ,[4]], [[] ,[1] ,[4, 3, 2]], [[] ,[2] ,[4, 3, 1]]
len(OPEN)=8; len(CLOSED)=53; COUNT = 53
OPEN is now: [[3] ,[4, 2] ,[1]], [[2] ,[4, 3, 1] ,[]], [[2] ,[4, 3] ,[1]], [[3, 1] ,[2] ,[4]], [[3, 2] ,[1] ,[4]], [[] ,
[1] ,[4, 3, 2]], [[] ,[2] ,[4, 3, 1]], [[3, 1] ,[4] ,[2]]
len(OPEN)=8; len(CLOSED)=54; COUNT = 54
OPEN is now: [[2] ,[4, 3, 1] ,[]], [[2] ,[4, 3] ,[1]], [[3, 1] ,[2] ,[4]], [[3, 2] ,[1] ,[4]], [[] ,[1] ,[4, 3, 2]], [[] ,
[2] ,[4, 3, 1]], [[3, 1] ,[4] ,[2]], [[3, 2] ,[4] ,[1]]
len(OPEN)=8; len(CLOSED)=55; COUNT = 55
OPEN is now: [[2] ,[4, 3] ,[1]], [[3, 1] ,[2] ,[4]], [[3, 2] ,[1] ,[4]], [[] ,[1] ,[4, 3, 2]], [[] ,[2] ,[4, 3, 1]], [[3,
1] ,[4] ,[2]], [[3, 2] ,[4] ,[1]], [[] ,[4, 3, 1] ,[2]]
len(OPEN)=8; len(CLOSED)=56; COUNT = 56
OPEN is now: [[3, 1] ,[2] ,[4]], [[3, 2] ,[1] ,[4]], [[] ,[1] ,[4, 3, 2]], [[] ,[2] ,[4, 3, 1]], [[3, 1] ,[4] ,[2]], [[3,
2] ,[4] ,[1]], [[] ,[4, 3, 1] ,[2]], [[] ,[4, 3, 2] ,[1]]
len(OPEN)=8; len(CLOSED)=57; COUNT = 57
OPEN is now: [[3, 2] ,[1] ,[4]], [[] ,[1] ,[4, 3, 2]], [[] ,[2] ,[4, 3, 1]], [[3, 1] ,[4] ,[2]], [[3, 2] ,[4] ,[1]], [[] ,
[4, 3, 1] ,[2]], [[] ,[4, 3, 2] ,[1]], [[3] ,[2, 1] ,[4]], [[3] ,[2] ,[4, 1]]
len(OPEN)=9; len(CLOSED)=58; COUNT = 58
OPEN is now: [[] ,[1] ,[4, 3, 2]], [[] ,[2] ,[4, 3, 1]], [[3, 1] ,[4] ,[2]], [[3, 2] ,[4] ,[1]], [[] ,[4, 3, 1] ,[2]], [[] ,
[4, 3, 2] ,[1]], [[3] ,[2, 1] ,[4]], [[3] ,[2] ,[4, 1]], [[3, 2, 1] ,[] ,[4]], [[3, 2] ,[] ,[4, 1]]
len(OPEN)=10; len(CLOSED)=59; COUNT = 59
OPEN is now: [[] ,[2] ,[4, 3, 1]], [[3, 1] ,[4] ,[2]], [[3, 2] ,[4] ,[1]], [[] ,[4, 3, 1] ,[2]], [[] ,[4, 3, 2] ,[1]],
[[3] ,[2, 1] ,[4]], [[3] ,[2] ,[4, 1]], [[3, 2, 1] ,[] ,[4]], [[3, 2] ,[] ,[4, 1]], [[1] ,[] ,[4, 3, 2]], [[] ,[] ,[4, 3, 2, 1]]
len(OPEN)=11; len(CLOSED)=60; COUNT = 60
OPEN is now: [[3, 1] ,[4] ,[2]], [[3, 2] ,[4] ,[1]], [[] ,[4, 3, 1] ,[2]], [[] ,[4, 3, 2] ,[1]], [[3] ,[2, 1] ,[4]], [[3] ,
[2] ,[4, 1]], [[3, 2, 1] ,[] ,[4]], [[3, 2] ,[] ,[4, 1]], [[1] ,[] ,[4, 3, 2]], [[] ,[] ,[4, 3, 2, 1]], [[1] ,[2] ,[4, 3]], [[] ,[2,
1] ,[4, 3]]
len(OPEN)=12; len(CLOSED)=61; COUNT = 61

OPEN is now: [[3, 2] ,[4] ,[1]], [[] ,[4, 3, 1] ,[2]], [[] ,[4, 3, 2] ,[1]], [[3] ,[2, 1] ,[4]], [[3] ,[2] ,[4, 1]], [[3, 2, 1] ,[] ,[4]], [[3, 2] ,[] ,[4, 1]], [[1] ,[] ,[4, 3, 2]], [[] ,[] ,[4, 3, 2, 1]], [[1] ,[2] ,[4, 3]], [[] ,[2, 1] ,[4, 3]], [[3] ,[4, 1] ,[2]], [[3] ,[4] ,[2, 1]]
len(OPEN)=13; len(CLOSED)=62; COUNT = 62
OPEN is now: [[] ,[4, 3, 1] ,[2]], [[] ,[4, 3, 2] ,[1]], [[3] ,[2, 1] ,[4]], [[3] ,[2] ,[4, 1]], [[3, 2, 1] ,[] ,[4]], [[3, 2] ,[] ,[4, 1]], [[1] ,[] ,[4, 3, 2]], [[] ,[] ,[4, 3, 2, 1]], [[1] ,[2] ,[4, 3]], [[] ,[2, 1] ,[4, 3]], [[3] ,[4, 1] ,[2]], [[3] ,[4] ,[2, 1]], [[3, 2, 1] ,[4] ,[]], [[3, 2] ,[4, 1] ,[]]
len(OPEN)=14; len(CLOSED)=63; COUNT = 63
OPEN is now: [[] ,[4, 3, 2] ,[1]], [[3] ,[2, 1] ,[4]], [[3] ,[2] ,[4, 1]], [[3, 2, 1] ,[] ,[4]], [[3, 2] ,[] ,[4, 1]], [[1] ,[] ,[4, 3, 2]], [[] ,[] ,[4, 3, 2, 1]], [[1] ,[2] ,[4, 3]], [[] ,[2, 1] ,[4, 3]], [[3] ,[4, 1] ,[2]], [[3] ,[4] ,[2, 1]], [[3, 2, 1] ,[4] ,[]], [[3, 2] ,[4, 1] ,[]], [[1] ,[4, 3] ,[2]], [[] ,[4, 3] ,[2, 1]]
len(OPEN)=15; len(CLOSED)=64; COUNT = 64
OPEN is now: [[3] ,[2, 1] ,[4]], [[3] ,[2] ,[4, 1]], [[3, 2, 1] ,[] ,[4]], [[3, 2] ,[] ,[4, 1]], [[1] ,[] ,[4, 3, 2]], [[] ,[] ,[4, 3, 2, 1]], [[1] ,[2] ,[4, 3]], [[] ,[2, 1] ,[4, 3]], [[3] ,[4, 1] ,[2]], [[3] ,[4] ,[2, 1]], [[3, 2, 1] ,[4] ,[]], [[3, 2] ,[4, 1] ,[]], [[1] ,[4, 3] ,[2]], [[] ,[4, 3] ,[2, 1]], [[1] ,[4, 3, 2] ,[]], [[] ,[4, 3, 2, 1] ,[]]
len(OPEN)=16; len(CLOSED)=65; COUNT = 65
OPEN is now: [[3] ,[2] ,[4, 1]], [[3, 2, 1] ,[] ,[4]], [[3, 2] ,[] ,[4, 1]], [[1] ,[] ,[4, 3, 2]], [[] ,[] ,[4, 3, 2, 1]], [[1] ,[2] ,[4, 3]], [[] ,[2, 1] ,[4, 3]], [[3] ,[4, 1] ,[2]], [[3] ,[4] ,[2, 1]], [[3, 2, 1] ,[4] ,[]], [[3, 2] ,[4, 1] ,[]], [[1] ,[4, 3] ,[2]], [[] ,[4, 3] ,[2, 1]], [[1] ,[4, 3, 2] ,[]], [[] ,[4, 3, 2, 1] ,[]]
len(OPEN)=15; len(CLOSED)=66; COUNT = 66
OPEN is now: [[3, 2, 1] ,[] ,[4]], [[3, 2] ,[] ,[4, 1]], [[1] ,[] ,[4, 3, 2]], [[] ,[] ,[4, 3, 2, 1]], [[1] ,[2] ,[4, 3]], [[] ,[2, 1] ,[4, 3]], [[3] ,[4, 1] ,[2]], [[3] ,[4] ,[2, 1]], [[3, 2, 1] ,[4] ,[]], [[3, 2] ,[4, 1] ,[]], [[1] ,[4, 3] ,[2]], [[] ,[4, 3] ,[2, 1]], [[1] ,[4, 3, 2] ,[]], [[] ,[4, 3, 2, 1] ,[]]
len(OPEN)=14; len(CLOSED)=67; COUNT = 67
OPEN is now: [[3, 2] ,[] ,[4, 1]], [[1] ,[] ,[4, 3, 2]], [[] ,[] ,[4, 3, 2, 1]], [[1] ,[2] ,[4, 3]], [[] ,[2, 1] ,[4, 3]], [[3] ,[4, 1] ,[2]], [[3] ,[4] ,[2, 1]], [[3, 2, 1] ,[4] ,[]], [[3, 2] ,[4, 1] ,[]], [[1] ,[4, 3] ,[2]], [[] ,[4, 3] ,[2, 1]], [[1] ,[4, 3, 2] ,[]], [[] ,[4, 3, 2, 1] ,[]]
len(OPEN)=13; len(CLOSED)=68; COUNT = 68
OPEN is now: [[1] ,[] ,[4, 3, 2]], [[] ,[] ,[4, 3, 2, 1]], [[1] ,[2] ,[4, 3]], [[] ,[2, 1] ,[4, 3]], [[3] ,[4, 1] ,[2]], [[3] ,[4] ,[2, 1]], [[3, 2, 1] ,[4] ,[]], [[3, 2] ,[4, 1] ,[]], [[1] ,[4, 3] ,[2]], [[] ,[4, 3] ,[2, 1]], [[1] ,[4, 3, 2] ,[]], [[] ,[4, 3, 2, 1] ,[]]
len(OPEN)=12; len(CLOSED)=69; COUNT = 69
OPEN is now: [[] ,[] ,[4, 3, 2, 1]], [[1] ,[2] ,[4, 3]], [[] ,[2, 1] ,[4, 3]], [[3] ,[4, 1] ,[2]], [[3] ,[4] ,[2, 1]], [[3, 2, 1] ,[4] ,[]], [[3, 2] ,[4, 1] ,[]], [[1] ,[4, 3] ,[2]], [[] ,[4, 3] ,[2, 1]], [[1] ,[4, 3, 2] ,[]], [[] ,[4, 3, 2, 1] ,[]]
len(OPEN)=11; len(CLOSED)=70; COUNT = 70
The Tower Transport is Triumphant!
Solution path:
[[4, 3, 2, 1] ,[] ,[]]
[[4, 3, 2] ,[1] ,[]]
[[4, 3] ,[1] ,[2]]
[[4, 3, 1] ,[] ,[2]]
[[4, 3] ,[] ,[2, 1]]
[[4] ,[3] ,[2, 1]]
[[4, 1] ,[3] ,[2]]
[[4, 1] ,[3, 2] ,[]]
[[4] ,[3, 2, 1] ,[]]
[[] ,[3, 2, 1] ,[4]]
[[1] ,[3, 2] ,[4]]
[[] ,[3, 2] ,[4, 1]]
[[2] ,[3] ,[4, 1]]
[[2, 1] ,[3] ,[4]]
[[2, 1] ,[] ,[4, 3]]
[[2] ,[1] ,[4, 3]]
[[] ,[1] ,[4, 3, 2]]

[[1] ,[] ,[4, 3, 2]]
[[] ,[] ,[4, 3, 2, 1]]
Length of solution path found: 18 edges
70 states expanded.
MAX_OPEN_LENGTH = 16