The Missionary Vs Cannibals problem is working for various scenarios like:

1. When Missionary and cannibals are 3 each:   
   Please enter # of Missionaries:

3

Please enter # of Cannibals:

3

Solution (Missionaries On Left Side, Cannibals On Left Side, Boat Current Direction, Missionaries On Right Side, Cannibal On Right Side):

(3, 3, L, 0, 0) -> (3, 1, R, 0, 2) -> (3, 2, L, 0, 1) -> (3, 0, R, 0, 3) -> (3, 1, L, 0, 2) -> (1, 1, R, 2, 2) -> (2, 2, L, 1, 1) -> (0, 2, R, 3, 1) -> (0, 3, L, 3, 0) -> (0, 1, R, 3, 2) -> (1, 1, L, 2, 2) -> (0, 0, R, 3, 3)

Depth of tree: 11

1. When Missionary are 4 and Cannibals are 3:   
   Please enter # of Missionaries:

4

Please enter # of Cannibals:

3

Solution (Missionaries On Left Side, Cannibals On Left Side, Boat Current Direction, Missionaries On Right Side, Cannibal On Right Side):

(4, 3, L, 0, 0) -> (4, 1, R, 0, 2) -> (4, 2, L, 0, 1) -> (2, 2, R, 2, 1) -> (3, 2, L, 1, 1) -> (2, 1, R, 2, 2) -> (2, 2, L, 2, 1) -> (0, 2, R, 4, 1) -> (0, 3, L, 4, 0) -> (0, 1, R, 4, 2) -> (1, 1, L, 3, 2) -> (0, 0, R, 4, 3)

Depth of tree: 11

1. When Missionary and Cannibal are 4 each: (I have thrown an instance of class Exception with the string message of “No solution found”)  
   Please enter # of Missionaries:

4

Please enter # of Cannibals:

4

Exception in thread "main" java.lang.Exception: [Error] no solution found. No solution possible when the Missionaries and Cannibals are equal to 4

At com.usu.MissionaryCannibalProblem.MissionaryAndCannibals.main(MissionaryAndCannibals.java:23)

1. When Missionary are 5 and Cannibals are 4:  
   Please enter # of Missionaries:

5

Please enter # of Cannibals:

4

Solution (Missionaries On Left Side, Cannibals On Left Side, Boat Current Direction, Missionaries On Right Side, Cannibal On Right Side):

(5, 4, L, 0, 0) -> (5, 2, R, 0, 2) -> (5, 3, L, 0, 1) -> (3, 3, R, 2, 1) -> (4, 3, L, 1, 1) -> (3, 2, R, 2, 2) -> (3, 3, L, 2, 1) -> (2, 2, R, 3, 2) -> (3, 2, L, 2, 2) -> (2, 1, R, 3, 3) -> (2, 2, L, 3, 2) -> (0, 2, R, 5, 2) -> (0, 3, L, 5, 1) -> (0, 1, R, 5, 3) -> (1, 1, L, 4, 3) -> (0, 0, R, 5, 4)

Depth of tree: 15