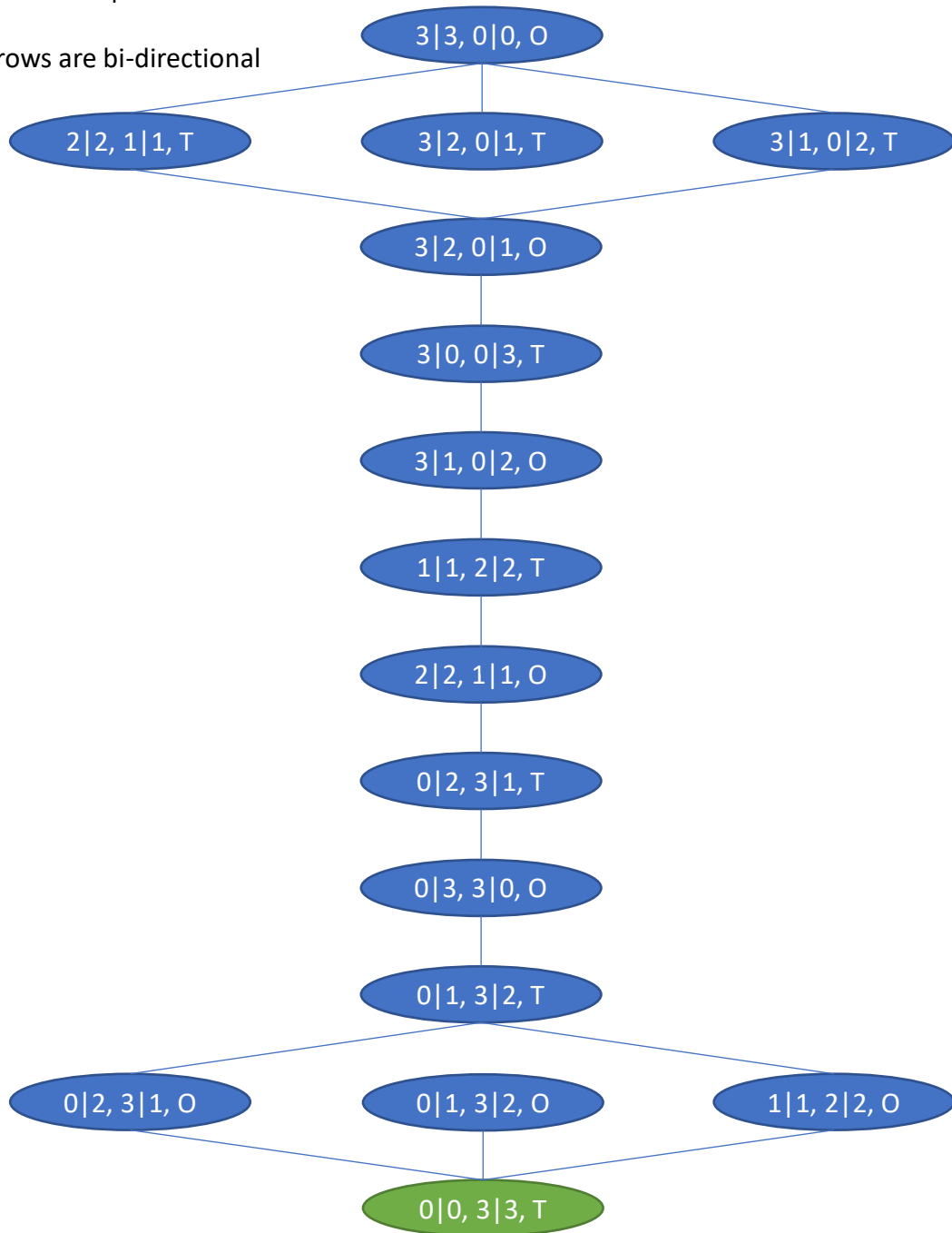


3a.

Legend:

Origin, Target, Boat Pos  
Missionaries | Cannibals

All arrows are bi-directional



I chose three algorithms: breadth-first, depth-first, and A\*. Due to the nominally linear shape of the state-space (16 nodes total) a breadth-first search would explore almost every node in the state space (15 explored nodes) and depth first would find a solution that doesn't explore any nodes not on the path from start to goal (12 explored nodes). A\* was chosen as an exercise to express that optimality does not always equate to minimum explored nodes (15 explored nodes).

3b. It is a good idea to check for repeated states. All edges in the graph are bi-directional so it is conceivable for a model to get stuck in an infinite loop without checking for duplicates. However, due to the small state space this can be ignored for smarter search algorithms like A\* which take the full path traveled into account.

## Program Output:

Begin breadth-first

Checking (3|3, 0|0, O)

Checking (2|2, 1|1, T)

Checking (3|2, 0|1, T)

Checking (3|1, 0|2, T)

Checking (3|2, 0|1, O)

Checking (3|0, 0|3, T)

Checking (3|1, 0|2, O)

Checking (1|1, 2|2, T)

Checking (2|2, 1|1, O)

Checking (0|2, 3|1, T)

Checking (0|3, 3|0, O)

Checking (0|1, 3|2, T)

Checking (0|2, 3|1, O)

Checking (1|1, 2|2, O)

Checking (0|0, 3|3, T)

Found goal node (0|0, 3|3, T)

15 nodes checked

Final path: [(3|3, 0|0, O), (2|2, 1|1, T), (3|2, 0|1, O), (3|0, 0|3, T), (3|1, 0|2, O), (1|1, 2|2, T), (2|2, 1|1, O), (0|2, 3|1, T), (0|3, 3|0, O), (0|1, 3|2, T), (0|2, 3|1, O), (0|0, 3|3, T)]

End breadth-first

-----  
Begin depth-first

Checking (3|3, 0|0, O)

Checking (3|1, 0|2, T)

Checking (3|2, 0|1, O)

Checking (3|0, 0|3, T)

Checking (3|1, 0|2, O)

Checking (1|1, 2|2, T)

Checking (2|2, 1|1, O)

Checking (0|2, 3|1, T)

Checking (0|3, 3|0, O)

Checking (0|1, 3|2, T)

Checking (1|1, 2|2, O)

Checking (0|0, 3|3, T)

Found goal node (0|0, 3|3, T)

12 nodes checked

Final path: [(3|3, 0|0, O), (3|1, 0|2, T), (3|2, 0|1, O), (3|0, 0|3, T), (3|1, 0|2, O), (1|1, 2|2, T), (2|2, 1|1, O), (0|2, 3|1, T), (0|3, 3|0, O), (0|1, 3|2, T), (1|1, 2|2, O), (0|0, 3|3, T)]

End depth-first

-----  
Begin A\*

Checking (3|3, 0|0, O): p = 0: h = 3.0

Checking (2|2, 1|1, T): p = 1: h = 2.0

Checking (3|1, 0|2, T): p = 1: h = 2.0

Checking (3|2, 0|1, T): p = 1: h = 2.5

Checking (3|2, 0|1, O): p = 2: h = 2.5

Checking (3|0, 0|3, T): p = 3: h = 1.5

Checking (3|1, 0|2, O): p = 4: h = 2.0

Checking (1|1, 2|2, T): p = 5: h = 1.0

Checking (2|2, 1|1, O): p = 6: h = 2.0

Checking (0|2, 3|1, T): p = 7: h = 1.0

Checking (0|3, 3|0, O): p = 8: h = 1.5

Checking (0|1, 3|2, T): p = 9: h = 0.5

Checking (0|2, 3|1, O): p = 10: h = 1.0

Checking (1|1, 2|2, O): p = 10: h = 1.0

Checking (0|0, 3|3, T): p = 11: h = 0.0

Found goal node (0|0, 3|3, T)

15 nodes checked

Final path: [(3|3, 0|0, O), (2|2, 1|1, T), (3|2, 0|1, O), (3|0, 0|3, T), (3|1, 0|2, O), (1|1, 2|2, T), (2|2, 1|1, O), (0|2, 3|1, T), (0|3, 3|0, O), (0|1, 3|2, T), (0|2, 3|1, O), (0|0, 3|3, T)]

End A\*