**Moore Traversal**

**A-Star Heuristic**

To compare the two states, the algorithm takes int account two aspects. One is the distance already covered through the swamp. The distance is calculated by counting dry patches of land passed till the current state. The other one is the distance still to be covered to reach the end of the swamp. This satisfies the equation: f(n) = h(n) + g(n), where h(n) is the estimated distance to the right-end if the swamp from the current state and g(n) is the actual distance already covered from the left-end to the current state. So, the sum of h(n) and g(n) yields the actual distance from left to the far right-end of the swamp. Thus, the heuristic of every state is going to be equal. Since their heuristic is the same, then the priority queue will sort them according to the First-In-First-Out rule.

**Performance comparison**

**DFS**

Based on my knowledge, DFS is always guaranteed to find the solution where there is one because it is a complete search algorithm. DFS is however not guaranteed to find the shortest path, hence it is not optimal. We follow the left-hand branches as far as possible and, If the solution is not found we back-trace to the parent state and choose a different path. So here it is highly possible that we end up visiting every node on the tree and that increases the cost which is something we literally don’t want.

**BFS**

Guaranteed to find the shortest path, however it might not be the most optimal. First iterates through the children states pushing them into a queue, moving along from left to right until all children are visited. It will stop when the goal state is reached and return the path to this state. In general, BFS is better than DFS and in this scenario BFS definitely is more optimal than DFS.