1. Develop a model of the environment. What constitutes a state of the environment? What is a successor state resulting of executing an action in a certain state? Which action is legal under which conditions? Maybe you can abstract from certain aspects of the environment to make the state space smaller.

The environment state contains the agent‘s current location, orientation, and home square along with the location of each obstacle and dirt spot, and whether the agent is turned on or off.

If the agent is off, the only legal action is TURN\_ON. If there‘s an obstacle in front of the agent, or the agent would move off the grid, OG is not a legal action. If the agent is on a square containing dirt, SUCK is the only legal action.

1. Estimate the size of the state space assuming the environment has width W, length L and D dirty spots.

The agent can face four different directions for each square it can occupy, and each state can have up to D remaining dirty spots. The state space is therefore (W\*L – o)\*D where o is the number of obstacle squares.

1. Assess the following blind search algorithms wrt. their completeness, optimality, space and time complexity in the given environment: Depth-First Search, Breadth-First Search, Uniform-Cost Search. If one of the algorithms is not complete, how could you fix it? Note: Do this step before you implement the algorithms, so you know what to expect when you run the algorithms. Otherwise you might be very surprised.

**DFS:** Not great. It is incomplete without a check for duplicate visited states,

and not optimal. Its time complexity of O(b^m) is massively worse than BFS in

the worst case, but its space complexity of O(bm) is significantly better than

that of BFS.

**BFS:** Some good, some bad. It's complete, but not optimal. Its time complexity of

O(b^d) is better than DFS, and its space complexity of O(b^d) is significantly

worse than DFS.

**Uniform Search:** Both complete and optimal. Its time and space complexity are O(b^(1+C/ε)).

Where b is the branching factor, m is the maximal depth of a leaf node, d is the depth of the depth of the shallowest goal node, C is the cost of the optimal solution, and ε is the minimum cost of an action.

1. Think of a good (admissible) heuristic function for estimating the remaining cost given an arbitrary state of the environment. Make sure, that your heuristics is really admissible!

(# of Dirts - 1)\*Min distance between dirts + Manhattan distance from Home to furthest dirt + Manhattan distance from agent to furthest dirt.