Question 1)

1. The state space is the set of all possible states the game can be in. Here, it is where 9 is the total number of squares and 2 is the number of possible states for each square. The state space includes 512 possible states. The initial state is the grid with all lights turned on. The goal state is the grid with all lights turned off.
2. Actions: *Actions (Lights-Out)* ∈ *{S1, S2, S3, S4, S5, S6, S7, S8, S9}* where *SX represents the squares that can be pressed.*

Transition model:

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |

*\*where ~ represents a binary switch in state, from lighted to non-lighted or vice versa*

1. A breadth-first graph search is guaranteed to find an optimal solution for this starting state. This is because of two reasons. One, the cost of each action is uniform, making breadth first search optimal, and two, graph search checks for redundant paths (checking states that have already been explored) before adding to the queue. Given that this state space involves many redundant paths, this is a necessary feature of an optimal search algorithm. Breadth first tree search does not check for redundant paths and will likely return a solution that involves one or more loops back onto a previous state, which would not be an optimal solution.
2. This heuristic function will expand the nodes within the frontier according to their number of lights switched on. It will expand the node with the minimum number of lights switched on. A priority queue is a data structure to arrange this order. This is a queue of nodes within the frontier ordered by the number of lights switched on. The first node in the queue, which is the node to be expanded, has the minimum number of lights switched on in the frontier.
3. This heuristic function is not admissible given that minimizing number of lights switched on and progressing towards terminal state (all lights switched off) are not equivalent in this game.

For instance: an optimal (5-step) solution to the problem from 9x9 state with fully lighted squares

Action (S1) 🡪 Action (S3) 🡪 Action (S7) 🡪 Action (S9) 🡪 Action(S5)

Compared to, the order of expanded nodes within this A\* search algorithm.

Action(S5) 🡪 Action(S1) 🡪 Action (S2) 🡪 Action (S8) 🡪 results in grid with 1 lighted square

In this game, a state may have a low number of lights switched on, but this necessarily does not reflect a proximity towards the terminal state. In other words, an optimal solution requires the player to accept an increase in the number of lights switched on in current state compared to previous state to achieve a winning terminal state. Hence the heuristic function leads the search towards nodes which are farther from the solution, rather than closer to it, so it does not improve efficiency of search.

1. We can define this problem as a CSP:

Set of variables:

Domains:

Constraints: – all squares must be off

1. A diagram of a network

   Description automatically generated with medium confidenceThe hypergraph of this problem is as follows.
2. Our global constraint is which can be broken down into binary constraints to be fed into the CSP solver.

One feature of the game is that states of S1, S4 and S7 cannot be affected by presses to S3, S6 or S9. Hence the first column should be completely off before presses to the third column.

Furthermore, the same logic applies to the first and last rows.

Combining these two constraints, the CSP checks S1,S2,S3,S4,S7 to satisfy the conditions to press S5,S6,S8,S9. Finally, we add the third global constraint.

Question 2

1. Detailed computation
2. The Bayes nets depicts all conditional dependencies. Since I know I don’t have Coronix, the probability of having a fever is condition only by having the flu. Since having a headache is not conditional by the flu (but only by Coronix) then the headache is independent of the flu.
3. Detailed as follows