COMP30024 Assignment

Duc Huy Le - 1254871
Xuan Bach Phan - 1166353

Question 1:

These are our step of A* implementation

- Calculate heuristic for all red nodes and put into priority queue
- Pop starting node from priority queue
- Get neighbors of the starting node
- Iterate through neighbors to find the optimal move.
- If the neighbor is the blue node, prefer this move. If not use heuristic to find optimal move
- Put next move into priority queue
- Repeat from step 2 until there are no blue nodes on the board

Choice of data structure: we choose priority queue to store the starting red nodes and the optimal next moves, the priority is the heuristic.

Let n and m be the number of reds and blues on the original board, k is the minimum steps needed to win the game.
Space complexity: O(n)
Time complexity:

- Cost of “linear_distance” and “distance” is O(1)
- Cost of “heuristic” is O(m)
- Cost of “expand” is O(1)
- Cost of putting element in priority queue using heap is O(log n)
- Cost of “search” is (n*m*k*log(n))

Question 2:

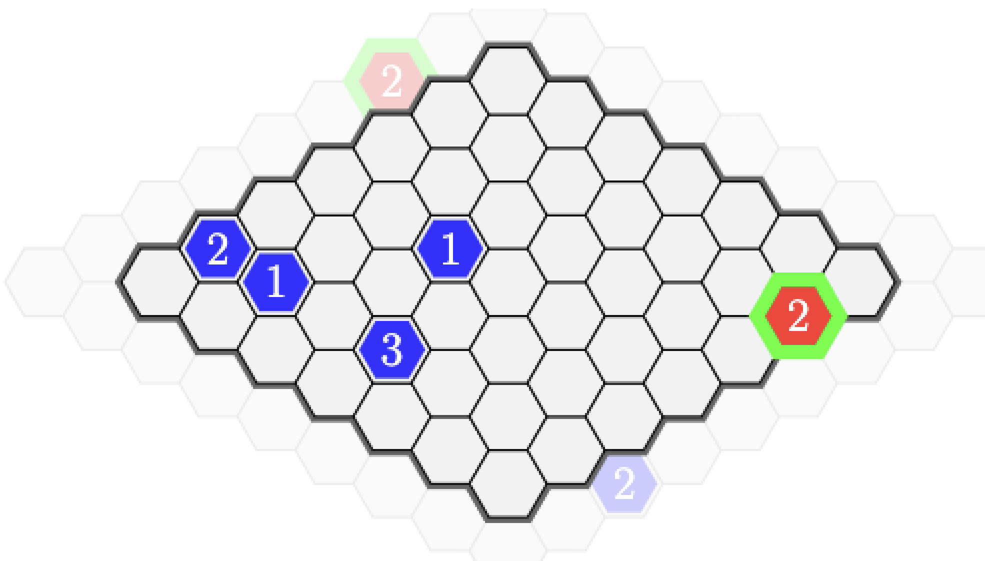
In solving the single Infection game, one approach is to use the average distance between the start point and the remaining blue points on the board to calculate the heuristic. This approach limits each move to having only one option, which can simplify the search function. However, the drawback of this heuristic is that it may not generate optimal solutions for every possible board.

For instance, in a board with a BLUE cell with power of 6, the solution generated using this heuristic may not be optimal. The termination of a cell occurs when it exceeds the limit power of 6. However, in the search function, the termination of a cell with excessive power is avoided. As a result, the search function tends to eliminate the BLUE cell with power of 6 in the last step, as this is the most efficient approach. However, this approach may not always optimize the solution, as it may not be possible to eliminate the BLUE cell with power of 6 in the middle of the search function which is the most optimal solution. In such cases, other heuristics or search algorithms may be needed to find the most optimal solution.

Overall, while the average distance heuristic can be useful in simplifying the search function for some boards in the single Infection game, it may not always generate optimal solutions, and alternative approaches may be necessary for more complex boards.

Question 3:

In the single Infecxion game, optimizing the search for the optimal solution requires consideration of both SPAWN and SPREAD actions. In contrast to generating neighbors for each loop, the search function should focus on identifying the most optimal SPAWN solution by looping through the neighbor cells of BLUE cells. By adding new RED to the neighbor cell targeted for SPREAD, the total number of steps required to win the game can be reduced. This approach requires careful consideration of the optimal SPAWN function, which can be determined by looping through the neighbor cells of BLUE cells as well.



For instance, the BLUE cell with the highest power on the table can be identified as the first cell to be targeted for SPREAD, leading to a reduction of one step in the total number of steps needed to win the game.

Therefore, to improve the efficiency and effectiveness of the search function in the single Infection game, it is essential to consider both SPAWN and SPREAD actions and aim for the most optimal SPAWN solution by looping through the neighbor cells of BLUE cells.