

1. We adopted the A* search algorithm. The specific implementation process is as follows: We implemented a priority queue (open_set) using heapq, and sorted according to the heuristic function to prioritize the expansion of the most promising nodes.

g_score is a dictionary used to record the actual movement cost from the starting point to each node.

came_from is also a dictionary used to record the parent node of each node and its movement action. It also records the backtracking path.

We also wrote two functions to handle the movement of the red frog, namely handle step move and handle jump move. Both of them update the path data uniformly through the Astar function. This process terminates when accessing any target grid, and then the final route is generated through backtracking.

Time complexity: $O(b^d)$ b represents the possible directions that the red frog can move. d indicates the number of steps for the shortest path.

Space complexity: $O(n)$ n represents the number of nodes for exploration.

2. Our heuristic calculation is the minimum value of Chebyshev distance, which is the maximum difference between the current coordinate and all targets in the minimum and maximum directions.

The Chebyshev distance conforms to the movement rule of frogs and ensures optimality. It does not overestimate the actual number of steps taken by the frog. The frog reduces at least one unit of Chebyshev distance with each step. By prioritizing the expansion of nodes closer to the target, the search speed is accelerated.

3.If six frogs are moved simultaneously, the problem will transform from a single-path planning to a multi-agent path planning. The complexity will increase significantly. Secondly, conflict issues need to be handled. The movement routes of different frogs may conflict with each other. For our current move function, conflict detection is necessary. We need to check whether the target square is occupied by other red frogs. Secondly, we can plan paths separately for each red frog, then compare whether the paths conflict with each other, and modify the routes to resolve the conflicts.