Roomba in a Maze

Kolmogorov group members

October 23, 2024

1 Original Version

Problem 1. Imagine that a roomba is placed in a rectangular maze where grid cells may be separated by horizontal or vertical walls.

You can remotely control the roomba by ordering it to move to a neighboring cell in one of four cardinal directions. If the roomba is given a command to move into a cell that is separated from the current one by a wall, the roomba bumps into the wall and remains in the current cell.

You know the layout of the maze. The maze is guaranteed to be connected, i.e., every cell can be reached from any other cell via a sequence of moves in cardinal directions that does not cross any walls.

However, the starting location of the roomba is unknown. The remote control has no feedback, i.e., there is no way to know if the roomba bumps into a wall or successfully moves to a neighboring cell as a result of any given command at any point in time.

The goal is to determine if there exists a finite series of commands such that the roomba ends up in the top left corner of the maze at the end, no matter its starting position.

To solve this problem, we will first perform a series of reductions to equivalent problems.

Problem 2. Determine if there exists a finite series of commands after which the location of the roomba becomes explicitly known, regardless of its starting position.

Indeed, the sequence of moves leading the roomba to the top left corner from any starting position is a solution for the variant problem. Conversely, if there is a sequence of moves after which we know the roomba's position, we can then move the roomba to the top left corner, since the maze is connected.

Problem 3. A roomba is placed in every cell of the maze. The goal is to determine if there is a sequence of moves that collects all the rombas in the same cell.

The equivalence follows from the fact that the unknown starting position of the roomba can be modeled by randomly picking one of the roombas starting in every cell of the maze.

Problem 4. Two roombas are placed in two given cells of the maze. The goal is to determine if there is a sequence of commands that moves both of these rombas to the same cell.

Clearly, a procedure that collects all roombas from all possible starting positions in one cell successfully solves the problem for any two roombas with fixed starting positions. Now suppose that for any two cells there is a sequence of commands that moves two roombas starting in those two cells to the same cell. Then we can derive a procedure for collecting the roombas from all starting locations as follows:

1. Keep track of the current set of distinct positions of roombas, starting with all cells of the maze.
2. If the set consists of only one location, then we are done. Otherwise pick any two different positions from the set.
3. Obtain a sequence of commands to move the roombas from these two starting positions to the same cell.
4. Update the current set of positions by applying the corresponding moves to all positions in the set.
5. Go to step 2.

Note that if two roombas share a cell, then they end up in the same cell after any move command. Thus, a sequence of commands can only reduce the size of the set of all distinct roomba positions. Moreover, the sequence chosen in step 3 merges two distinct positions into one, so our set becomes smaller by at least one element after step 4. Thus, if we can solve the problem for two roombas starting in any two given cells, then we can solve the problem for roombas starting in all cells of the maze.

Now let us solve the the problem for two roombas. Let us label the roombas with numbers 1 and 2 and repeatedly perform the following steps.

1. If the roombas are in the same cell, we are done.
2. Otherwise find a shortest path from the current location of roomba 1 to the current location of roomba 2. Issue the sequence of commands moving roomba 1 along this path from start to finish.
3. Go to step 1.

First we show that the sequence from step 2 cannot increase the shortest path length between the roombas. Note that roomba 1 does not bump into a walls at any point in this sequence by construction. Now consider roomba 2. On the one hand, if a command makes it bump into a wall, then the shortest path length between roombas 1 and 2 strictly decreased as a result of this move. On the other hand, if a command successfully moves roomba 2 to a neighboring cell, then there is a path between roombas 1 and 2 that has the same length as in the configuration just before this command. Thus, in all cases the shortest path length between the roombas did not increase.

Next, observe that if the shortest path length between the roombas stayed the same after the sequence of moves from step 2, then neither roomba bumped into a wall at any point. Indeed, this holds for roomba 1 by construction, and if roomba 2 bumps into a wall at any point, then the shortest path length strictly decreases.

Now, note that in the case where all commands resulted in a valid move for both roombas, the positions of both roombas shifted closer to the boundary. More formally, assume that the starting column of roomba 2 is strictly to the left of that of roomba 1 (up to rotation of the maze), and assume that the sequence of moves from step 2 resulted in no wall bumps for roomba 2. Then after that command sequence roomba 2 ends up in a column strictly to the left of its starting column. Since the maze is finite, roomba 2 will bump into a wall after a finite number of iterations of our algorithm — otherwise we can repeat our argument and conclude that roomba 2 eventually ends up in a column out of bounds of the maze. Therefore, after a finite number of iterations of our algorithm the shortest path length between the roombas will strictly decrease as a result of a wall bump by roomba 2. And since the shortest path length is finite and is bounded from below by 0, which is our exit condition, our algorithm will terminate after finitely many repetitions of the main loop.

Thus, a composition of the previously described algorithms produces a finite sequence of commands that moves the two roombas to the same cell.

Observe that once the roombas from all possible starting positions have been moved to the same cell, we can move them all together to an arbitrary cell in the maze.

2 Extension to Multiple Mazes

2.1 Two Mazes

Problem 5. Imagine that there are two layouts of the maze that are connected and have the same size, and the roomba is placed in a random cell of a randomly selected layout. Is there a sequence of moves after which the roomba ends up in the top left corner, no matter its starting position and no matter the maze layout?

Observe that performing sequences of moves from the original problem for each of the two mazes consecutively reduces this problem to the following.

Problem 6. Suppose there are two layouts of the maze that are connected and have the same size. Suppose the roomba is placed either in a given cell in the first maze or in another given cell in the second maze. Is there a sequence of moves after which the roomba ends up in the top left corner in either case?

For the sake of simplicity, we will refer to the configurations of the roomba's position in the corresponding maze as roomba 1 and roomba 2. Consider the following procedure.

1. Perform the sequence of commands moving roomba 1 from its current position to the top left corner. Update the position of roomba 2 accordingly.

2. If roomba 2 is in the top left corner, we are done.

3. Otherwise issue a sequence of commands that moves roomba 2 from its current position to the top left corner along a shortest path. Update the position of roomba 1 accordingly.

4. Swap labels of roombas 1 and 2.

5. Go to step 2.

Claim. After each iteration, the shortest path length from the current position of roomba 2 to the top left corner (in the corresponding maze) strictly decreases.

Proof. Let ℓ and ℓ' denote the shortest path length before and after the sequence of commands from step 3, respectively.

First, note that $\ell' \leq \ell$. Indeed, by construction the sequence from step 3 contains ℓ commands. When issued to roomba 1, some of these commands may result in wall bumps, so there is a path of length $\leq \ell$ between the top left corner and the new position of roomba 1, and the shortest path length could be even smaller. Thus, after the relabeling in step 4 we get $\ell' \leq \ell$.

Now let us show that $\ell' < \ell$. To this end, observe that $\ell' = \ell$ is only possible if roomba 1 never bumps into a wall when it is issued the commands from step 3. Since these commands move roomba 2 from its position to the top left corner without wall bumps, the horizontal and vertical position of roomba 1 shifts by the same amount as for roomba 2 after this sequence of moves. However, the position of roomba 2 shifts up and to the left by a non-zero amount (up to rotating and mirroring the mazes, we may assume that roomba 2 moves some columns to the left). This leads to a contradiction, as the same shift for roomba 1 would leave it out of bounds of the maze. □

To complete the proof, note that the shortest path length is finite and bounded from below by 0. Thus, after a finite number of iterations roomba 2 ends up in the top left corner together with roomba 1 and we are done.

Observe that it is not always possible to move both roombas to the same arbitrary cell in both mazes. For example, suppose that the target cell is located at the end of a long dead end that must be entered from the left in one maze and from the right in the other maze. Then the only position from which it is reachable is itself, so it is impossible to move both roombas to that cell simultaneously from any other pair of starting positions.

2.2 Three Mazes

Problem 7. Imagine that there are three layouts of the maze that are connected and have the same size, and the roomba is placed in a random cell of a randomly selected layout. Is there a sequence of moves after which the roomba ends up in the top left corner, no matter its starting position and no matter the maze layout?

solution or counter example?

2.3 All Connected Mazes of Same Size

Problem 8. Suppose we only know the size of the maze and that it is connected, but the layout of the maze is unknown. Is there a sequence of moves after which the roomba ends up in the top left corner, no matter its starting position and no matter the maze layout?

solution or counter example?

3 Extension to Labeled Digraphs

Let $k \in \mathbb{Z}_{\geq 1}$. Let $D = (V, A)$ be a digraph with vertex set V and (labeled) arc set A satisfying the following conditions:

- for every vertex v, there are k outgoing arcs $a_{v,1}, \ldots, a_{v,k} \in \delta^+(v) \subseteq A$, some of which may be loops or may lead ;
- D is strongly connected, i.e., for any pair of vertices $v, u \in V$ there exists a directed path from v to u along arcs from A and vice versa;
-

  analog of boundary condition for rectangular mazes? no looping move sequences?

4 Other Extensions

- Minimize number of moves
- Unknown maze size: 2 options, range, set
- Game: players take turns to move the roomba, the goal is to end in the top-left corner, repeating positions is not allowed