If we assume that the room can be divided into arrays/matrices we can use one of the following graph-based algorithms:

Breadth First search:

Starts at a root position (0,0 in our case) and visits all nodes at the current depth level before moving on to the nodes at the next depth level. We can divide the nodes to visited nodes and unvisited nodes to avoid visiting the same node more than once.

Depth First Search:

Starts at a root position (0,0 in our case) and visits all nodes at the current depth level before moving on to the nodes at the next depth level. We can divide the graph into arrays then divide them to visited arrays and unvisited arrays to avoid visiting the same array more than once.

A*

…….

SLAM algorithms

Kalman filter Based SLAM:

This is a recursive algorithm that works in two steps: a prediction step and an update step. The prediction step makes a prediction for the state based on the dynamics of the system, and the update steps reconcile the prediction with the measurements to produce an improved estimate of the state.

Graph based SLAM:

Graph based SLAM treats the SLAM problem as a graph problem, where position information is represented by the nodes and the map is derived with the edges. In the robotics space, it's common to use pose-graph optimization. In a pose graph, the nodes represent poses and landmarks and the edges represent constraints between them. New nodes are added as new poses and landmarks are detected, and constraints connect sequential nodes with information about the movement.

As far as I know proper SLAM for robotics is done using lidar sensors with Imu and encoders and it has a ROS plugin that manages those sensors and integrates them together.