# Project 2 Answers

## Q1

### a)

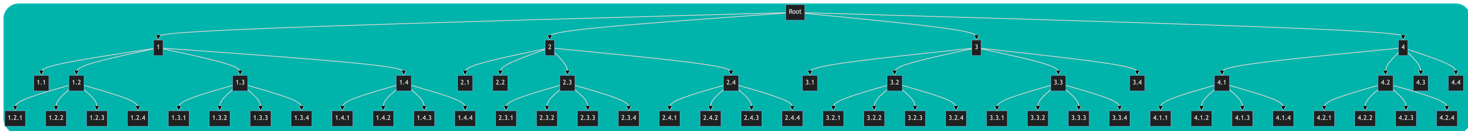
* One problem with using a uniform grid for path planning is that all cells have the same size, which can lead to a lot of unnecessary cells being created. This can lead to a lot of wasted memory and computation time. This happens when the resolution of the grid is too high, i.e. smaller cell sizes.
* On the contrary, when the resolution is too low, i.e. larger cell sizes, the path planning algorithm may not be able to find a path around the obstacle, since the whole cell is considered as an obstacle, where in reality the robot might be able to pass through the empty part of the cell.
* Also, if the resolution is too low, the path may not be smooth and may have sharp turns.
* The pros of increasing the grid resolution are:
  + More accurate path planning.
  + Smoother paths.
  + Better environment representation in real-world scenarios.
  + Better obstacle avoidance.
  + The cons of increasing the grid resolution are:
  + More memory usage.
  + More computation time.

### b)

For the quadtree, we are assuming the numbering of the quads is clockwise starting from the top left. That is, the top left quad is 1, the top right is 2, the bottom right is 3, and the bottom left is 4. For the tree structure, node 3.1.4 means 3rd quad in the first level, 1st quad in the second quad, and 4th quad in the third level. Note: sorry, we could not make the font in the tree structure larger as the tree is too wide.

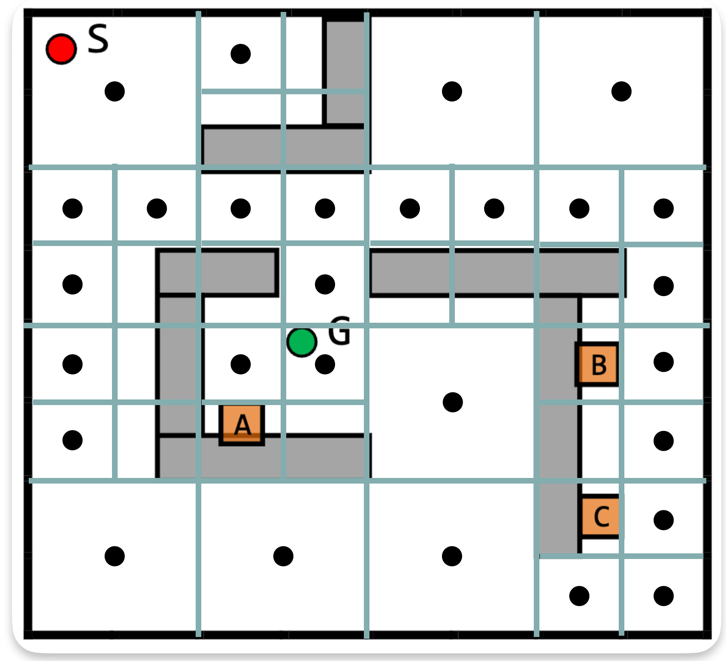
A grid with a diagram

Description automatically generated with medium confidence

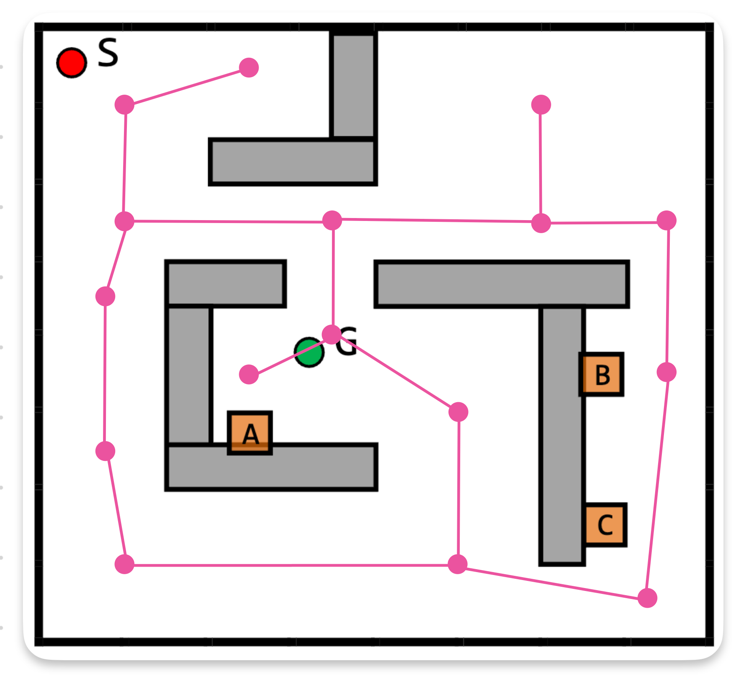


### c)

Below are the generated waypoints.



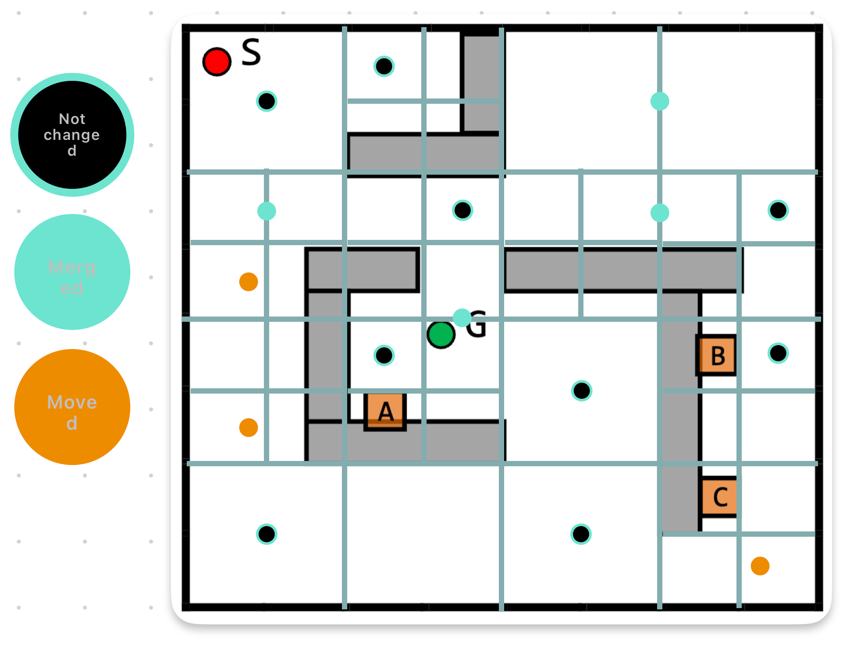
Below is the optimized topological map of the connected map. The number of waypoints was reduced from 28 before optimizing to only 16 after optimizing.



Below is a color guide of the waypoints that were moved, merged, or unchanged.

We followed the same philosophy to merge *any* two waypoints.: waypoints are merged as long as the merging does not compromise the robot’s ability to reach all areas. E.g.: the cyan dot in the left side of the upper half of the map. No need for two separate dots to reach any other waypoint on the map.

We slightly moved some waypoints to allow a smoother turning of the robot and to make the robot evenly spaced from obstacles as much as possible. E.g., the two orange dots on the left. If they were not moved to the right, the robot would have been too close to the left wall.



### d)

The below map takes the Start (S) and Goal (G) waypoints into consideration. Only three new edges are added to the graph (shown in blue), one for the S node and two for the G node. Everything else on the map is left as is.

