Overall Idea: Connect nodes (phone poses) that are connected by floor (downward projections of the poses hit the same horizontal plane, the line segment between the poses does not hit a vertical plane, and the line segment also does not exit the polygon of the floor plane).

Idea 1

- Create map as normal
- Before data is uploaded + processed
    - Use stored plane data to ray trace each node vertically downwards to find common horizontal planes
    - Trace between nodes that share horizontal planes to check for distance, walls and exiting the plane polygon
    - Note nodes that share a floor within a certain distance, without going through a wall or leaving the plane
- Note: each node in the JSON must store the IDs of all nodes that are connected to it rather than just its own ID
    - i.e. graph instead of linked list
- Perform optimization on server, ignoring the extra connections
- Create map based on optimized poses with extra connections
- Pros:
    - Minimal restructuring of data and code
- Cons:
    - Relies on minimal drift (not gonna happen)

Idea 2

- Create map as normal
    - While collecting data, raytrace straight down and also around
    - Store pose to the anchor of the horizontal plane(s) it hits
    - Store pose to the anchor of the first vertical plane it hits in each direction (ignoring repeats)
- Store planes in JSON
    - Pose of the anchor point
    - Transform of each connected node to the anchor point
    - Boundary vertices
- Add planes to G2O optimization with stored poses and transforms
- Optimize plane positions with everything else
    - Make sure to strongly weight plane orientation!
- If two planes of ~same orientation overlap based on optimized anchor positions and boundary vertices, combine them (taking the union of the sets of nodes that connect to each)
    - If one plane is completely contained within another, delete it

- Run connection algorithm based on optimized/combined planes
- Create map with appropriate pose connections