Se necesita programar la lógica para un autoelevador (“zamping”) robótico, cuya tarea consiste en recolectar y llevar hacia un punto de entrega único, una determinada cantidad de pallets que le son pedidos. El robot conoce la ubicación de los pallets dentro del almacén, que se encuentran siempre en el suelo (no hay estanterías ni pallets apilados), conoce su propia ubicación, la ubicación del punto de entrega (fijo), y recibe como orden una lista de números de pallets a recolectar.

Podemos pensar al almacén como una grilla para modelar los movimientos del robot como discretos. Y de la misma manera, podemos considerar el “agarrar” un pallet, y “dejar” un pallet, como movimientos atómicos, sin necesidad de conocer el detalle de cómo se efectúan dichas tareas. Para agarrar un pallet, el robot debe estar en la posición del pallet, y para dejarlo, debe encontrarse en el punto de entrega.

El aspecto que se desea optimizar es la cantidad de movimientos que el robot realiza para llevar todos los pallets pedidos, al punto de entrega. Y para realizar el ejemplo de árbol de búsqueda utilizaremos el almacén de ejemplo diagramado a la derecha.

Resolver:

1. Defina formalmente el problema para ser resuelto mediante búsqueda. Escriba código Python y donde considere necesario agregue comentarios que esclarezcan el funcionamiento del código.
2. Plantee y programe una heurística admisible para ser utilizada en este problema. (no importa la calidad de la misma, pero debe ser admisible)
3. Teniendo en cuenta el diagrama de almacén, y como orden “entregar los pallets 8, 3 y 9”, resuelva mediante búsqueda A\* considerando la heurística planteada en b (solo las primeras 5 iteraciones).

