Lab Notes: Graphs, Paths & Search

Objective

The goal of this lab is to understand and utilise various search algorithms (Depth First Search, Breadth First Search, Dijkstra's, and A* Search) in a simple "box" based world.

Modifying Searches

1.Edge Cost Value: In the `box_world.py` module, the `min_edge_cost` variable for the A* algorithm is set incorrectly. Changing this value affects the optimality of the algorithms search.

2. Diagonal Edges: In the `reset_navgraph` method of the `BoxWorld` class (in `box_world.py`), This code modifies the navigation graph to include diagonal movements.

Algorithms:

- BFS: Explores nodes level by level. It may not be efficient for large maps due to the fact that it explores a significant amount of nodes.
- DFS: Explores nodes along a path until a dead end is reached, then backtracks. It can be less efficient and can get stuck in loops.
- Dijkstra's: Finds the shortest path from the start to the target by exploring the lowest-cost path so far. It is optimal but can be slower due to exploring all possible paths.
- A-Star: Combines the path cost so far and the estimated cost to the target, the heuristic must be admissible.