# CMPSC 441: Artificial Intelligence (Fall 2025) [Homework-3]

1. **A* Search - Delivery Robot at Penn State Harrisburg Eastgate Center**

You are designing an autonomous delivery robot for the Penn State Harrisburg Eastgate Center, a downtown Harrisburg facility hosting state agencies like the Pennsylvania Securities Commission. The robot must navigate a grid-based office layout to deliver documents from a reception desk to an office while avoiding obstacles like desks and walls. Use A* search to find the shortest path, combining actual distance traveled (g) with a heuristic estimate (Manhattan distance, h) to the goal. This scenario connects to Penn State Harrisburg's role in fostering innovation and industry partnerships, such as those in technology and AI, and reflects a practical application in a professional setting near the campus.

To add depth, you must implement diagonal movements (8 directions) if allowed, compute custom heuristics for different terrains (e.g., some cells are "slow" zones with higher cost), and handle dynamic obstacles by rerunning A* if needed. This requires thinking about heuristic admissibility and path optimization.

**Instructions**

a) Create a file named eastgate_astar.py for your implementation.
b) At the top of eastgate_astar.py, include the line: from eastgate_helper import get_office_grid, get_terrain_costs, test_scenarios to import the grid, costs, and testing function.
c) Implement A* with support for 8 directions (including diagonals; cost 1 for ortho, $\sqrt{2} \approx 1.414$ for diagonal).
d) Use a custom heuristic that accounts for terrain costs (e.g., Manhattan adjusted by average cost).
e) Incorporate terrain costs from get_terrain_costs(): some cells have higher g-costs (e.g., 2 for "slow" zones like carpeted areas).
f) Your function should return the shortest path (list of (row, col) tuples) considering costs, or None if blocked.
g) Use the imported test_scenarios to test with multiple start-goal pairs and dynamic changes (e.g., adding an obstacle mid-path).
h) Think about: Why must the heuristic be admissible? How do diagonal moves affect optimality?
i) Extra Credits: Implement a visualization function to print the grid with path and costs.

**Submission**:

- **(50 points) Solution**: python file (s).
- **(50 points) Report**: That contains –
  - Brief description of the problem with your solution approach. **[15 points]**
  - Provide output solutions with brief explanatory notes (screenshots). **[20 points]**
  - Answer/solutions parts (h) and (i) **[15 points]**
  - Mention/discuss any assumptions that you made during the process.