CS156 In-Class Exercise #1: SEARCH

GROUP MEMBER NAMES:

1. Alisha Rath
2. Rashmi Vishwanath Bhat
3. Mohit Kundrl
4. Farhan Ansari

ENTER “COMPLETE” OR “INCOMPLETE” here \_\_\_\_”COMPLETE”\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# INTRODUCTION:

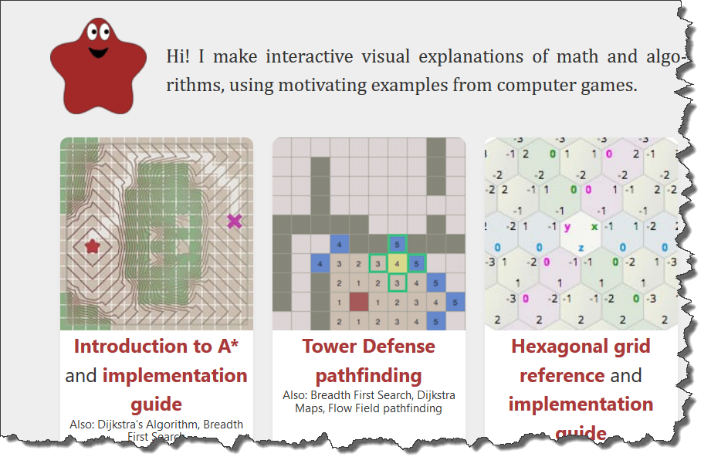
In this In-Class Exercise, you will get hands on experience with various search algorithms that were presented in class, and in particular the A\* algorithm.

# INSTRUCTIONS:

1. Launch your favorite browser and go to https://www.redblobgames.com/

from Red Blob Games.

Then click on the link to take you to a demonstration of the A\* algorithm, or simply use the following URL: https://www.redblobgames.com/pathfinding/a-star/introduction.html



NOTE: The emphasis of this demo is on A\*, but it covers several other search algorithms such as breadth first and uniform cost search first as a lead up to A\*. A\* might not have been covered completely in lecture before this in-class exercise, BUT it will be covered by the time this exercise is due to be submitted to Canvas.

1. Read the entire article and follow the instructions for each of the different search algorithms and diagrams. Throughout the article you will see buttons like  . Make sure that you move your mouse to hover over the button to see the change/point of the article.
2. Make sure that you go through each animation.
3. When you get to the section entitled  you can download the code form the Code folder. Download the files named Search.py and implementation.py. Use the tutorial to help you understand the code. The tutorial takes you step by step through the various search algorithms and how they work.
4. In the file named implementation.py, there is a function named heuristic\_2 that needs to be completed. Implement an “As The Crow Flies” heuristic for heuristic\_2. Then in the function a\_star\_search, change the call to heuristic\_1 to heuristic\_2. Then run the A\* algorithm with heuristic\_2 (i.e., the “As The Crow Flies” heuristic) and compare the result, e.g., cost, with that of heuristic\_1 (“The Manhattan Distance” heuristic).
5. Answer as many of the questions below, if any, and upload your answers to the Canvas In-Class 1 Assignment. If you did not reach the point where you can answer any questions below, then take screen snapshots of your progress. Include them in this MS Word document at the bottom, and then upload the document to the Canvas In-Class 1 Assignment.

# QUESTIONS:

1. Which heuristic, in step 5, performed better?

Answer: A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

came\_from: {(1, 4): None, (2, 4): (1, 4), (1, 3): (1, 4), (0, 4): (1, 4), (1, 5): (1, 4), (1, 6): (1, 5), (0, 5): (1, 5), (2, 5): (1, 5), (2, 6): (1, 6), (0, 6): (1, 6), (2, 3): (2, 4), (3, 4): (2, 4), (3, 5): (2, 5), (3, 6): (2, 6), (4, 6): (3, 6), (0, 3): (0, 4), (0, 7): (0, 6), (0, 8): (0, 7), (0, 9): (0, 8), (1, 2): (1, 3), (3, 3): (2, 3), (2, 2): (2, 3), (3, 2): (3, 3), (4, 3): (3, 3), (0, 2): (0, 3), (1, 9): (0, 9), (1, 1): (1, 2), (2, 9): (1, 9), (2, 1): (2, 2), (3, 9): (2, 9), (4, 2): (3, 2), (3, 1): (3, 2), (4, 4): (3, 4), (4, 5): (3, 5), (4, 9): (3, 9), (4, 7): (4, 6), (5, 6): (4, 6), (5, 9): (4, 9), (4, 8): (4, 9), (5, 8): (5, 9), (6, 9): (5, 9), (7, 9): (6, 9), (6, 8): (6, 9), (6, 7): (6, 8), (7, 8): (6, 8)},

cost\_so\_far: {(1, 4): 0, (2, 4): 1, (1, 3): 1, (0, 4): 1, (1, 5): 1, (1, 6): 2, (0, 5): 2, (2, 5): 2, (2, 6): 3, (0, 6): 3, (2, 3): 2, (3, 4): 6, (3, 5): 7, (3, 6): 4, (4, 6): 9, (0, 3): 2, (0, 7): 4, (0, 8): 5, (0, 9): 6, (1, 2): 2, (3, 3): 3, (2, 2): 3, (3, 2): 4, (4, 3): 8, (0, 2): 3, (1, 9): 7, (1, 1): 3, (2, 9): 8, (2, 1): 4, (3, 9): 9, (4, 2): 9, (3, 1): 5, (4, 4): 11, (4, 5): 12, (4, 9): 10, (4, 7): 14, (5, 6): 14, (5, 9): 11, (4, 8): 15, (5, 8): 16, (6, 9): 12, (7, 9): 13, (6, 8): 13, (6, 7): 18, (7, 8): 14}

heuristic \_2

A screenshot of a computer

Description automatically generated

came\_from: {(1, 4): None, (2, 4): (1, 4), (1, 3): (1, 4), (0, 4): (1, 4), (1, 5): (1, 4), (2, 5): (2, 4), (2, 3): (2, 4), (3, 4): (2, 4), (1, 6): (1, 5), (0, 5): (1, 5), (3, 5): (2, 5), (2, 6): (2, 5), (0, 6): (1, 6), (3, 6): (2, 6), (4, 6): (3, 6), (0, 3): (1, 3), (1, 2): (1, 3), (3, 3): (2, 3), (2, 2): (2, 3), (3, 2): (3, 3), (4, 3): (3, 3), (0, 7): (0, 6), (1, 1): (1, 2), (0, 2): (1, 2), (2, 1): (2, 2), (0, 8): (0, 7), (4, 2): (3, 2), (3, 1): (3, 2), (4, 4): (3, 4), (0, 9): (0, 8), (4, 5): (3, 5), (0, 1): (0, 2), (1, 0): (1, 1), (2, 0): (2, 1), (4, 7): (4, 6), (5, 6): (4, 6), (3, 0): (3, 1), (4, 1): (3, 1), (1, 9): (0, 9), (2, 9): (1, 9), (3, 9): (2, 9), (4, 9): (3, 9), (5, 9): (4, 9), (4, 8): (4, 9), (5, 8): (5, 9), (6, 9): (5, 9), (7, 9): (6, 9), (6, 8): (6, 9), (5, 3): (4, 3), (0, 0): (0, 1), (6, 7): (6, 8), (7, 8): (6, 8)},

cost\_so\_far: {(1, 4): 0, (2, 4): 1, (1, 3): 1, (0, 4): 1, (1, 5): 1, (2, 5): 2, (2, 3): 2, (3, 4): 6, (1, 6): 2, (0, 5): 2, (3, 5): 7, (2, 6): 3, (0, 6): 3, (3, 6): 4, (4, 6): 9, (0, 3): 2, (1, 2): 2, (3, 3): 3, (2, 2): 3, (3, 2): 4, (4, 3): 8, (0, 7): 4, (1, 1): 3, (0, 2): 3, (2, 1): 4, (0, 8): 5, (4, 2): 9, (3, 1): 5, (4, 4): 11, (0, 9): 6, (4, 5): 12, (0, 1): 4, (1, 0): 4, (2, 0): 5, (4, 7): 14, (5, 6): 14, (3, 0): 6, (4, 1): 10, (1, 9): 7, (2, 9): 8, (3, 9): 9, (4, 9): 10, (5, 9): 11, (4, 8): 15, (5, 8): 16, (6, 9): 12, (7, 9): 13, (6, 8): 13, (5, 3): 13, (0, 0): 5, (6, 7): 18, (7, 8): 14}

Based on above we can see that more calculations are done for heuristic 2 than heuristics 1. H1 has an early exist but cost incurred in both the cases is the same that is 14.

1. Which heuristic is dominant?

Answer: In this scenario, where the total cost incurred is the same for both heuristics (14 in this case), Heuristic 1 is dominant because it will not expand more nodes than heuristic 2.

1. Are either or both optimal?

Answer: In the context of A\* search, "optimal" typically refers to finding the lowest-cost path from a start point to a goal point. A\* search, when using admissible heuristics, guarantees to find the optimal path, i.e., the lowest-cost path.

In the provided scenario where both heuristics result in the same total cost (14), it indicates that both heuristics have found an optimal path. A\* search, when using admissible heuristics, will always find the lowest-cost path if one exists.

Therefore, in this specific scenario, both heuristics have produced an optimal path in terms of total cost, as the total cost is the same for both

1. Is the grid best represented as a graph or a tree?

Answer: In the context of pathfinding algorithms like A\*, Dijkstra's algorithm, and breadth-first search, the grid is typically best represented as a graph.

Each cell in the grid corresponds to a node in the graph, and the edges between nodes represent valid movements between neighboring cells (up, down, left, right).

This representation allows for efficient traversal and pathfinding using graph search algorithms.

1. Based on your answer to 4, which criterion must the heuristic meet in order to be optimal?

Answer : In the context of a grid-based pathfinding problem, the actual cost from a node

n tothe goal is typically the minimum cost required to move from node

nto the goal using valid moves (e.g., up, down, left, right).

When a heuristic satisfies this admissibility criterion, A\* search using that heuristic will always find the lowest-cost path from the start node to the goal node. This is because the heuristic provides a lower bound on the actual cost to reach the goal, and A\* uses this information to guide the search efficiently while guaranteeing optimality.

Screenshots of progress:

Completed the path by hand:

A screenshot of a computer

Description automatically generated

Using heuristic 1 Manhattan distance A\*

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

came\_from: {(1, 4): None, (2, 4): (1, 4), (1, 3): (1, 4), (0, 4): (1, 4), (1, 5): (1, 4), (1, 6): (1, 5), (0, 5): (1, 5), (2, 5): (1, 5), (2, 6): (1, 6), (0, 6): (1, 6), (2, 3): (2, 4), (3, 4): (2, 4), (3, 5): (2, 5), (3, 6): (2, 6), (4, 6): (3, 6), (0, 3): (0, 4), (0, 7): (0, 6), (0, 8): (0, 7), (0, 9): (0, 8), (1, 2): (1, 3), (3, 3): (2, 3), (2, 2): (2, 3), (3, 2): (3, 3), (4, 3): (3, 3), (0, 2): (0, 3), (1, 9): (0, 9), (1, 1): (1, 2), (2, 9): (1, 9), (2, 1): (2, 2), (3, 9): (2, 9), (4, 2): (3, 2), (3, 1): (3, 2), (4, 4): (3, 4), (4, 5): (3, 5), (4, 9): (3, 9), (4, 7): (4, 6), (5, 6): (4, 6), (5, 9): (4, 9), (4, 8): (4, 9), (5, 8): (5, 9), (6, 9): (5, 9), (7, 9): (6, 9), (6, 8): (6, 9), (6, 7): (6, 8), (7, 8): (6, 8)},

cost\_so\_far: {(1, 4): 0, (2, 4): 1, (1, 3): 1, (0, 4): 1, (1, 5): 1, (1, 6): 2, (0, 5): 2, (2, 5): 2, (2, 6): 3, (0, 6): 3, (2, 3): 2, (3, 4): 6, (3, 5): 7, (3, 6): 4, (4, 6): 9, (0, 3): 2, (0, 7): 4, (0, 8): 5, (0, 9): 6, (1, 2): 2, (3, 3): 3, (2, 2): 3, (3, 2): 4, (4, 3): 8, (0, 2): 3, (1, 9): 7, (1, 1): 3, (2, 9): 8, (2, 1): 4, (3, 9): 9, (4, 2): 9, (3, 1): 5, (4, 4): 11, (4, 5): 12, (4, 9): 10, (4, 7): 14, (5, 6): 14, (5, 9): 11, (4, 8): 15, (5, 8): 16, (6, 9): 12, (7, 9): 13, (6, 8): 13, (6, 7): 18, (7, 8): 14}

heuristic \_2

A screenshot of a computer

Description automatically generated

came\_from: {(1, 4): None, (2, 4): (1, 4), (1, 3): (1, 4), (0, 4): (1, 4), (1, 5): (1, 4), (2, 5): (2, 4), (2, 3): (2, 4), (3, 4): (2, 4), (1, 6): (1, 5), (0, 5): (1, 5), (3, 5): (2, 5), (2, 6): (2, 5), (0, 6): (1, 6), (3, 6): (2, 6), (4, 6): (3, 6), (0, 3): (1, 3), (1, 2): (1, 3), (3, 3): (2, 3), (2, 2): (2, 3), (3, 2): (3, 3), (4, 3): (3, 3), (0, 7): (0, 6), (1, 1): (1, 2), (0, 2): (1, 2), (2, 1): (2, 2), (0, 8): (0, 7), (4, 2): (3, 2), (3, 1): (3, 2), (4, 4): (3, 4), (0, 9): (0, 8), (4, 5): (3, 5), (0, 1): (0, 2), (1, 0): (1, 1), (2, 0): (2, 1), (4, 7): (4, 6), (5, 6): (4, 6), (3, 0): (3, 1), (4, 1): (3, 1), (1, 9): (0, 9), (2, 9): (1, 9), (3, 9): (2, 9), (4, 9): (3, 9), (5, 9): (4, 9), (4, 8): (4, 9), (5, 8): (5, 9), (6, 9): (5, 9), (7, 9): (6, 9), (6, 8): (6, 9), (5, 3): (4, 3), (0, 0): (0, 1), (6, 7): (6, 8), (7, 8): (6, 8)},

cost\_so\_far: {(1, 4): 0, (2, 4): 1, (1, 3): 1, (0, 4): 1, (1, 5): 1, (2, 5): 2, (2, 3): 2, (3, 4): 6, (1, 6): 2, (0, 5): 2, (3, 5): 7, (2, 6): 3, (0, 6): 3, (3, 6): 4, (4, 6): 9, (0, 3): 2, (1, 2): 2, (3, 3): 3, (2, 2): 3, (3, 2): 4, (4, 3): 8, (0, 7): 4, (1, 1): 3, (0, 2): 3, (2, 1): 4, (0, 8): 5, (4, 2): 9, (3, 1): 5, (4, 4): 11, (0, 9): 6, (4, 5): 12, (0, 1): 4, (1, 0): 4, (2, 0): 5, (4, 7): 14, (5, 6): 14, (3, 0): 6, (4, 1): 10, (1, 9): 7, (2, 9): 8, (3, 9): 9, (4, 9): 10, (5, 9): 11, (4, 8): 15, (5, 8): 16, (6, 9): 12, (7, 9): 13, (6, 8): 13, (5, 3): 13, (0, 0): 5, (6, 7): 18, (7, 8): 14}