CSCI 3202
Lecture 8
September 10, 2025

Siri ... What do I do when confronted with a bear!? Run or play dead!?
Did you mean "dare"?
How the inventor of Siri deserves to meet his destiny
WUMO

Credit: Wumo by Wulff and Morgenstern. https://wumo.com

Announcements

- HW 2 due today by 11:59 pm
- HW 3 released today. Due Friday, Sept 19 by 11:59 pm
- Canvas won't regrade Quiz 2 automatically
  - Question which method uses a Queue
  - Answer is BFS, but quiz was not correct
  - I will regrade and release
- Quiz #3 on Friday
  - A*
  - Greedy
  - Heuristics

Homework 2

- Create a data structure to hold the graph. I would use a dictionary
- Define a graph_problem subclass of class Problem and fill in the needed functions.
- graph_problem should use a data structure that holds your graph
- In addition to `utils4e.py`, you will need to copy `utils.py` into your local

directory

- Visit the nodes in ascending order only when you have to choose between several nodes, such as when visiting the children of a node
  - You may need to sort parts of your data structure to make this work with the AIMA code
- Return the optimal path for BFS. The path found by DFS may or may not be optimal. Return the path found by DFS

Homework 3

- The obstacles in Problem are randomly assigned. They will change each time you rerun the problem *unless* you set the code to use the same seed each time it generates a new map
  - If you set the seed at the top of the code, then run the map generation separately, the maps will be different each time
  - If you set the seed in the map generation, the maps should be the same
- **You don't have to turn in a particular map for your homework**
- Some of the seeds create very dysfunctional maps. If you get one of these, rebuild the map with a different seed.

Lecture

- Heuristics
  - Not all heuristics are about distance
    Astar and Heuristics Slides.pdf
- Want to choose the largest, but still admissible heuristic
- Could we use h(n)= actual Euclidean distance to goal?

Search Questions

1. What does optimal mean for A*?
2. Why is a larger (but still admissible) heuristic better?
3. Can I create a new heuristic from existing ones?
   - Let h1(n) and h2(n) be admissible heuristics
   - Is h(n) = h1(n) + h2(n) admissible
   - What about h(n) = (h1(n) + h2(n)) / 2
   - if h1(n) ≤ h2(n)∀n, which heuristic, h1(n) or h2(n) will visit the fewest nodes to create the optimal path with A∗?

Algorithm Questions

1. Assume you have a UCS search method, but want to create a BFS search path. How can you do it? Is the resulting path optimal?
2. You are using an A* search method. If you set the heuristic to 0 for all nodes, will your result be optimal? Is this method consistent?
3. For the same A* method, what happens if you set the edge weights to 0 for all edges (but keep the heuristic)? Is the result optimal? Consistent?

Path

- Path construction for A* is the same as for UCS
- Start at the end and work backwards
  - Start at destination
  - Find the most recent node that *explored* the destination
  - Mark this node as being on path
  - Repeat with this node as destination
  - Done when you reach the starting node
- Often implemented as a Python dictionary
  - Each node is a key
  - Value is node that visited or explored it
  - If there are multiple explorations, dictionary only keeps most recent one

Next Class

- Wrap up on Searching
- Next week, begin Probability and Bayes Rule