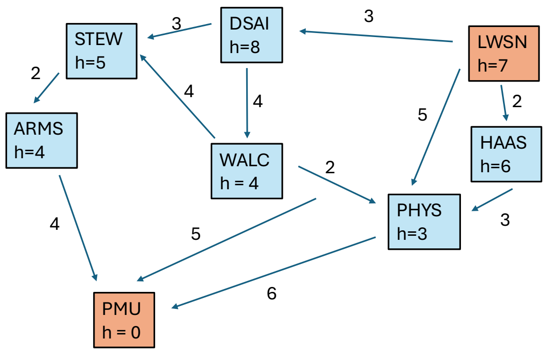
CS471 Assignment 1

Due date: Sunday Sep 21, 2025 (11:59pm Eastern Time)

This assignment includes both written exercises and a programming component. Please follow the submission instructions carefully. The "written" portion of the assignment must be typeset with LaTeX using the provided template.

Part 1: Written Assignment (40 pts)

1. (26 pts) Alice wants to go from the Lawson Computer Science Building (LWSN) to the Purdue Memorial Union (PMU). Please help Alice plan her route using the map illustrated below (the number shown along an edge represents the cost of going from the starting node to the ending node of that edge).



   (a) (20 pts) Please use each of the following search algorithms to plan the route for Alice. For each algorithm, write down (i) the order of the nodes expanded when executing the **tree** search version of the algorithm, and (ii) the order of the nodes that Alice actually will visit according to the trip you plan for her. As you plan the routes, if multiple options are available when expanding the next node, use **alphabetical order** for tie-breaking.

      i. (4pts) Breadth-first search
      ii. (4pts) Depth-first search
      iii. (4pts) Uniform cost search
      iv. (4pts) Greedy search using the values of h for each node as the heuristic values
      v. (4pts) A* search with the same heuristic

   (b) (6pts) Suppose Alice has a good friend Bob who has just finished an office hour session in STEW. After hearing that Alice plans to go to the Purdue Memorial Union (PMU), Bob invites Alice to stop by at STEW on her way to PMU. If Alice indeed comes to STEW, Bob will buy her a drink and go to PMU together with her. Let's assume that the possible get-together with Bob means a *reward* of 4 to Alice. So, we can slightly change the graph above by adding a new edge from STEW to PMU with a cost of -4 and plan the route again based on this new graph.

   i. Can we get an optimal route plan when solving this problem using the A* algorithms listed in Question 1(a)(v)? What is your reasoning?
   ii. What is the updated optimal trip plan?

2. (14pts) Admissible and consistent heuristics

   (a) (8pts) Prove that if a heuristic is consistent, it must be admissible. (Hint: A heuristic function $h(n)$ is consistent if the estimated cost $h(n)$ for any node n is no greater than the sum of the step cost from node n to any neighbor nodes n' ($n \rightarrow n'$) and the estimated cost $h(n')$ for node n'.)

   (b) (6pts) Construct a heuristic function that is admissible but not consistent on a small search graph. Show the search graph and explain why the heuristic function you construct is admissible but not consistent.

Submission

Please upload your answers to the **written** questions (i.e., Part 1) as a **pdf** in Gradescope:

- You should already have received an e-mail with the link to access Gradescope. If you haven't, let the TAs know.
- For your pdf file, use the naming convention `username_hw#.pdf`. For example, your TA with username *alice* would name her pdf file for HW1 as `alice_hw1.pdf`.
- Use the provided LaTeX template to typeset the assignment by editing the tex files under `student_response/`.
- After uploading your submission to Gradescope, mark each page to identify which question is answered on the page. (Gradescope will facilitate this.)
- Follow the above convention and instructions for future assignments as well.

Part 2: Programming Assignment (60 pts + 28 pts bonus)

For the programming assignments, we use the Pacman project designed for the course CS188 at UC Berkeley. Please see the detailed instructions in `project1.pdf` This classic assignment has been widely adopted at various institutes, e.g., UW, UIUC, UCLA, etc.

Please remember that solutions to any assignment should be your own. Using other people's solutions, within or outside Purdue goes against the academic honesty policy of this course. **The TAs will use code similarity measures to detect plagiarism cases against projects submitted in previous years, submitted by classmates, and code online (e.g., from Github), when grading the assignment.**

This homework uses Python 3, follow the Python tutorial in `project0.pdf` to set up the environment.

1. (60 pts) Complete Questions 1-4 described in `project1.pdf`. Submit your modified versions of `search.py` for grading. We will multiply your original score returned by `autograder.py` (12 pts in total) by 5.
2. (28pts bonus) Complete Questions 5-8 described in `project1.pdf`. Submit your modified versions of `searchAgents.py` for grading. We will multiply your original score returned by `autograder.py` (14 pts in total) by 2.

Submission

Please upload the following files: `search.py` and `searchAgents.py` to Gradescope.