CP468: Artificial Intelligence – Spring 2025

Assignment #1

Due on Feb 14th, 2025, at 11:50 PM

Note:

- The assignment's general instructions are posted in MyLS.
- Any updates or corrections will be posted on MyLS.
- Some the conceptual problems are from or inspired from the course textbook.
- Late submission will not be accepted.

Assignment Part I (Conceptual):

Problem #1 [10 marks]: For each activity, give a PEAS description of the task environment and characterize it in terms of the properties listed in Section 2.3.2 of the textbook.

- Playing soccer.
- Exploring the subsurface oceans of Titan.
- Shopping for used AI books on the Internet.
- Playing a tennis match.
- Practicing tennis against a wall.
- Performing a high jump.
- Knitting a sweater.
- Bidding on an item at an auction.

Problem #2 [10 marks]: For each of the following assertions, say whether it is true or false and support your answer with examples or counterexamples where appropriate.

a) An agent that senses complete information about the state is always perfectly rational.
b) There exist task environments in which a pure reflex agent can always behave rationally.
c) There exists a task environment in which no agent can be rational.
d) The input to the agent function is the same as the input to the agent program.

Problem #3 [14 marks]: You are designing an autonomous drone navigation system that operates in a hierarchical decision tree structure. The drone starts at an initial altitude level (state 1), and at each decision point (state k), it has two possible navigation choices: "Left" ($2k$) & "Right" ($2k + 1$)

a) Draw the portion of the navigation state space for states 1 to 15, representing the possible paths the drone could take.
b) Suppose the goal state is state 11, representing the target altitude level where the drone needs to stabilize. Determine the order in which states will be explored using:
    1. Breadth-First Search (BFS)
    2. Depth-Limited Search (DLS) with a limit of 3

3. Iterative Deepening Search (IDS)

c) How effectively can Greedy Best-First Search (GBFS) and A Search be applied to this problem? The heuristic function h(k) can be defined as the estimate of the remaining cost to reach state 11. A simple heuristic could be

$$h(k) = \lceil log2_2(11) \rceil - \lceil log_2(k) \rceil$$

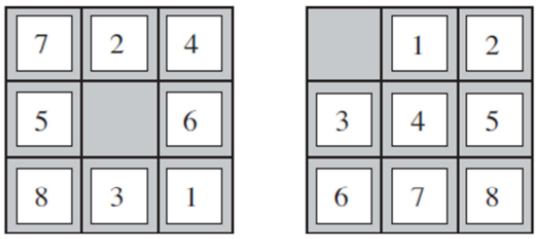
Compute and compare the GBFS and A* search order using this heuristic.

Problem #4 (Chapter 5, problem 9) [16 marks]: This problem exercises the basic game-playing concepts, using tic-tac-toe (noughts and crosses) as an example. We define X_n as the number of rows, columns, or diagonals with exactly n $X's$ and no $O's$. Similarly, O_n is the number of rows, columns, or diagonals with just n $O's$. The utility function assigns +1 to any position with X_3= 1 and −1 to any position with O_3 = 1. All other terminal positions have utility 0. For nonterminal positions, we use a linear evaluation function defined as $Eval(s) = 3X_2(s) + X_1(s) - (3O_2(s) + O_1(s))$.

a) Approximately how many possible games of tic-tac-toe are there?
b) Show the whole game tree, starting from an empty board and going down to depth 2 (i.e., one X and one O on the board), taking symmetry into account.
c) Mark the evaluations of all the positions at depth two on your tree.
d) Using the minimax algorithm, mark the backed-up values for the positions at depths 1 and 0 on your tree. Then, use those values to choose the best starting move.

Assignment Part II (Programming):

Note: Use the programming language of your choice.



Task (I) A*, 8-puzzle

Implement the A* algorithm for the 8-puzzle, with heuristics h_1 and h_2, as discussed in the course textbook. Generate randomly 100 reachable states of the 8-puzzle, solve them with h_1 and h_2 and record in a table the number of steps to find the solution and the number of nodes expanded by A* in each case. Find a third heuristic (h_3) for the 8-puzzle from the literature and repeat the

above to complete the table. Based on the table, comment on the performance of the three heuristics h_1, h_2, h_3.

Task (II) A*, 15-puzzle

Modify your code from task I to solve the 15-puzzle with heuristics h_1 and h_2, as discussed in the course textbook. Generate randomly 100 reachable states of the 15-puzzle, solve them with h_1 and h_2 and record in a table the number of steps to find the solution and the number of nodes expanded by A* in each case. Use the third heuristic (h_3) for the 15-puzzle, or find another third heuristic from the literature and repeat the above to complete the table. Based on the table, comment on the performance of the three heuristics h_1, h_2, h_3.

Table Template:
Use the following format to present your results:

Puzzle Type	Heuristic	Average Steps to Solution	Average Nodes Expanded
8-puzzle	h1		
8-puzzle	h2		
8-puzzle	h3		
15-puzzle	h1		
15-puzzle	h2		
15-puzzle	h3		

Submission Instructions

By the assignment deadline, one designated person from the group will need to upload the following to MLS:

- One designated person from each group should submit the assignment on MLS.
- A PDF report will be created for the first part of the assignment. Please ensure that the names and student IDs (SIDs) of all group members are listed on the first page of the report. The report file should be named in the format group#_a1.pdf based on your assigned group number.
- For the second part of the assignment, your source code should be compressed to a single **ZIP** file. The code archive should be named **group#_a1.zip** (based on the assigned group number).
- The code must be well commented, and it should be easy to see the correspondence between its contents and those in the report. Any supporting files (e.g., utility libraries) need to be included. If that will conclude a large Zip file, you must only provide a hyperlink from where it could be downloaded. You don't need to include any Bin or Executable file.

- **Include a README file explaining how to run the code**.
- Failing to excuse the code after the submission will be **penalized**.
- Multiple attempts will be allowed, but only your last submission before the deadline will be graded.
- We reserve the right to take off points for not following the above instructions.

Evaluation Criteria – Part II: Programming Tasks (100 Marks)

Component	Marks
Correct implementation of A* for task I	15
Correct implementation of A* for task II	15
Use of h1 and h2 heuristics (for the two tasks)	10
Selection and explanation of h3 (for the two tasks)	20
Tabulated results for 8-puzzle (for the two tasks)	10
Tabulated results for 15-puzzle (for the two tasks)	10
Analysis of heuristic performance	20

Notes:

- Ensure your code is efficient, especially for the 15-puzzle.
- Comment your code for readability.
- Cite any external resources used. This will include the use of any AI tool.

Good Luck!