CS 4820/5820 – Homework 1 (Fall 2025)

Problem Solving and Search

Please note that you can use any programming language of choice. Please do not use a graph library though. I want you to write the programs from scratch. You will also answer some theoretical questions and provide a writeup in AAAI format.

Instructions:

- Write clear and well-commented code (Python, Java, or another approved language).
- For each question, include: (a) your code/algorithm design (or pseudocode where applicable), (b) the output/result preferably in tables, and (c) a short explanation (2–3 sentences).
- Submit a single PDF containing your answers, explanations, and screenshots of code outputs using AAAI format with unnecessary template sections and subsections removed.
- Also submit your code for each question with appropriate description/manual/comments for how to run the code.
- Collaboration: You should work individually, no collaboration is allowed on homework assignments.
- Late submissions will not be accepted.

Part A: Intelligent Agents (1.0 pts)

1. Define the **PEAS** (Performance, Environment, Actuators, Sensors) description for the following AI systems:
   (a) An autonomous delivery drone.
   (b) A chess-playing agent.
   (c) A self-driving taxi.
2. Classify each system above as simple reflex, model-based reflex, goal-based, or utility-based agent, and explain why.

Part B: n-Puzzle problem (3.5 pts)

Consider the n-Puzzle problem where we have to place n tiles on an $n+1$ board by sliding them around so that the integers 1 through $n-1$ are in succession. $n+1$ must be a square integer. In particular,

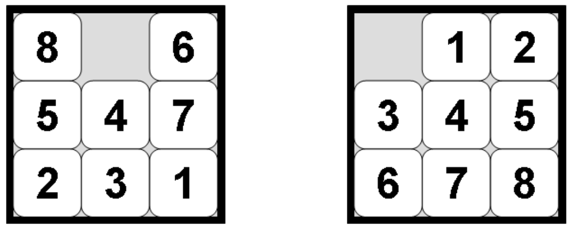

Figure 1: A random initial state and a goal state for the 8-puzzle problem

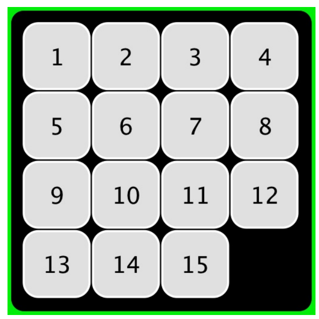

Figure 2: The goal state for the 15-puzzle problem

our text book discusses the 8-Puzzle problem (See Figure 1). See Figure 2 for a 15-size version. We can make it to be any size where $n = m^2 - 1$ for an integer $m > 1$.

Here is what is needed in this question.

- Define the problem formally (states, initial state, actions, transition model, goal test, path cost).
- Implement Breadth-First Search (BFS) and Depth-First Search (DFS), Iterative Deepening Search (IDS) and Bidirectional Search (BDS) for solving the n-puzzle problem with n=8 and 15 and compare the performance.
- Run each search method with each n (8 and 15) minimum of 3 times (3 runs) and report on their performance (average across runs and individual run execution times, average across runs and individual run's number of nodes expanded). each run is to use a random initial configuration.

Part C: n-queens Problem (4.0 pts)

We have discussed the n-queens problem in class. The goal is to place n-queens on a $n \times n$ chess board so that no queen attacks any other queen.

Here is what is needed in this question.

- Formulate the state space, initial state, actions, and goal test.

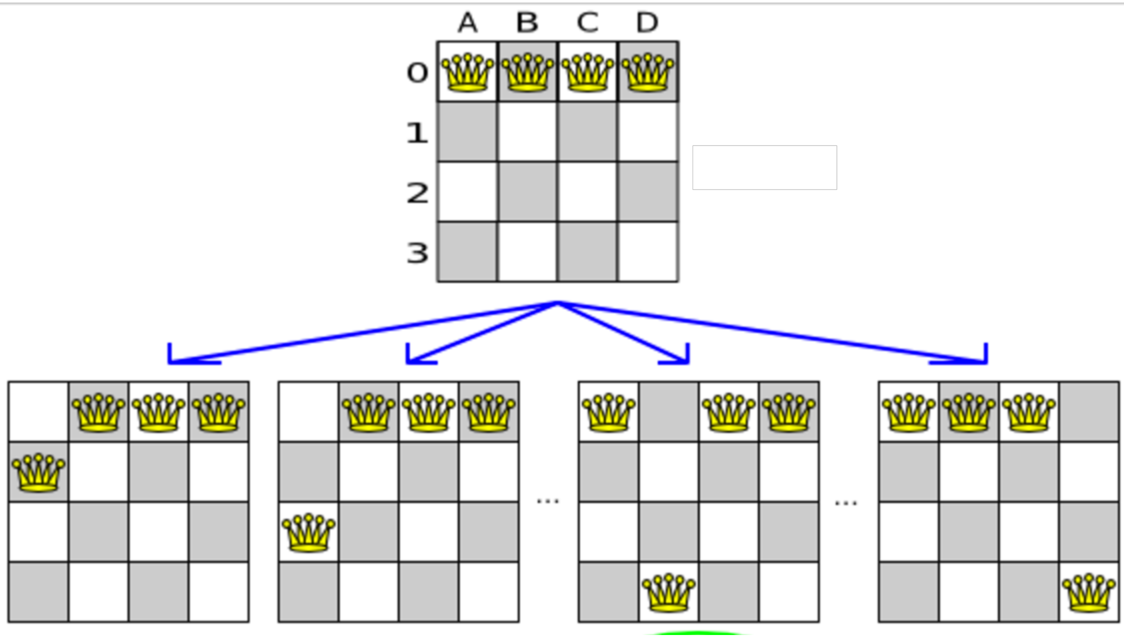

Figure 3: One level of graph for a 4-queens problem, for a generally used start state

- Implement both **Simulated Annealing (SA)** and a **Genetic Algorithm (GA)** for solving the n-queens problem with $n = 4$ and $n = 8$.
- For GA, experiment with different parameter settings (e.g., population sizes, crossover and mutation rates, maximum number of generations) and report how these choices affect performance. Try and report performance comparison on at least three completely different setting configurations with n=4 problem. Discuss the performance differences you observe and provide reasoning. Use the same initial state on these configurations and parameter setting tests.
- Run each method (SA and GA [use best configuration you found in previous point]) for both $n = 4$ and $n = 8$, with at least 3 runs per method. Use different random initial states for each run. Report and compare their performance in terms of:
  - Average and individual run execution times
  - Average and individual run number of nodes expanded
  - Quality of the solution found (e.g., number of conflicts if not solved optimally)
- Provide a comparison and discussion of the results, highlighting the impact of parameter choices on the effectiveness of SA and GA.

Part D: Heuristic Search (4.0 pts)

Implement A* search for the 8-puzzle using the following heuristics:

1. Misplaced tile heuristic.
2. Manhattan distance heuristic.

Compare their performance (number of nodes expanded, solution depth, execution time) on the same initial configuration.

For each heuristic above, state whether it is **admissible** and **consistent**, and justify your reasoning.

Part E: Advanced Challenge (Bonus +1.0 pt)

Solve the 15-puzzle using A* with Manhattan distance heuristic. Report the time, depth, and number of nodes expanded for 3 randomly initiated runs. Discuss why this problem is more computationally demanding than the 8-puzzle.

Note for Students: This assignment covers topics from lectures on Introduction to AI, Intelligent Agents, Problem Solving by Search, and Heuristic Search.