COMP-424: Artificial intelligence, Fall 2022

Homework 2

Due on *myCourses* Sunday Oct 30, 9:00pm.

General instructions.

- This is an individual assignment. You can discuss solutions with your classmates, but should only exchange information orally, or else if in writing through the discussion board on *Ed*. All other forms of written exchange are prohibited.
- Unless otherwise mentioned, the only sources you should need to answer these questions are your course notes, the textbook, and the links provided. Any other source used should be acknowledged with proper referencing style in your submitted solution.
- For Q1, you must code a solution as well as answering some questions in report form. For Q2 and Q3, you can solve manually, or write a program to help you. In all cases, can use a programming language of your choice. You can modify code from other sources if you provide adequate citation; this cannot be code from other students in the class.
- Submit one single pdf of your responses and one single code file for Q1, both through *myCourses*.

Question 1: Code N-Queens [40]

Write a program, in a language of your choice, that solves the N-Queens constraint-satisfaction problem using a Local Search method (Problem Definition on Lecture 3 PDF nodes, slides 46 and 47, Local Search details slides 87 and 88). You can use helper functions from the language you pick, and libraries if you find them helpful for the problem setup, but you must code the core AI solution logic yourself. Submit your code as one file and write about your findings in max 1 page of typed report in your PDF submission file. We will read your code but not run it – readability and commenting are important.

Use your code to answer the following required elements of the report:

a) Describe your solution in detail, using the language of Local Search. Do not copy your code in whole or in part. Rather give a concise intuitive explanation.

b) Print the solution of the largest N you managed as a list of N integers, zero-indexed and comma separated. C^{th} entry having value R means the queen in col C is in column R. (e.g., the solution for N=4 shown on slide 88 is [2, 0, 3, 1]). If you manage the normal chessboard size, N=8, there will be no further impact on grades (but whoever solves max N has the honor!)

c) What is the run-time complexity of your method, as a function of N? You can answer this by showing a table of N vs run-times or by attempting to write a big-oh expression. In either case, give an explanation referencing elements of your code solution.

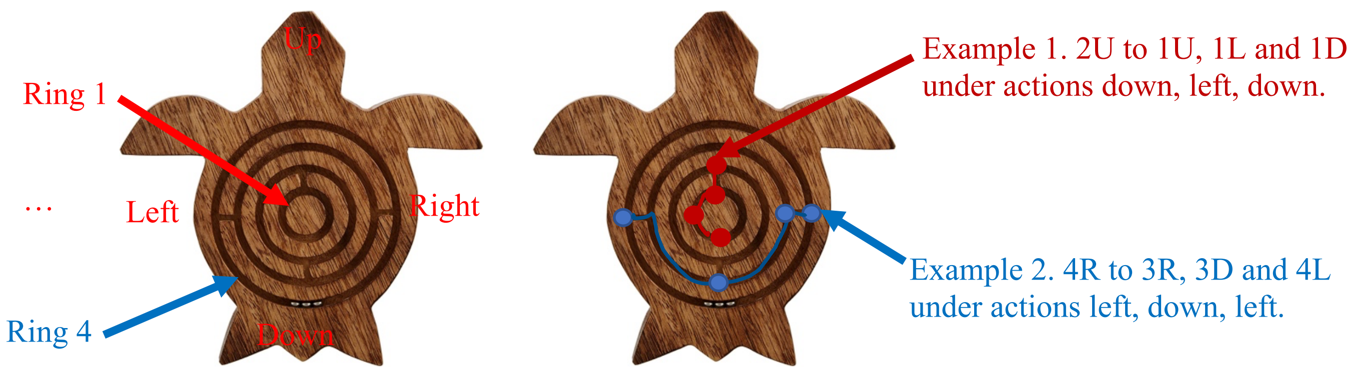
d) Did you encounter any local optima? Describe that situation and how you overcame it.

Hint: Our sample solution in Python is about 50 lines without comments. There are 4 kinds of constraints (rows, cols, diags parallel to y=x and diags parallel to y=-x). To code these:

- Use an array-like data structure with one int for each column, representing the row of each queen. This ensures we can never have two queens in the same column.
- Row constraints rule out duplicates in your array
- y=x diag constraints rule out pairs of queens sharing the same (row-col)
- y=-x diag constraints rule out pairs of queens sharing the same (row+col)

Question 2: Belief Space and Conformant Planning [30]

Consider an unobservable state planning problem where N balls can start anywhere within the puzzle shown here. Our job is getting all of the balls into Ring 1, regardless of where they start.



We will simplify the game to 4 critical positions per ring: up, down, left and right. Four actions, with the same names mean we tip the puzzle in that direction and hold it until all balls come to rest. Ignore ball collisions. Balls roll along the rings, resting when they either balance stably on top of a "hill" or at the bottom of a "cup". Balls pass through any "gap" they touch at any point in their path if the current action is aligned with the gap. Examples:

1. One ball starting at 2U passes the gap with action down, resting on the hill at 1U. After left it rolls into the cup at 1L. Finally, after down it ends at 1D.
2. One ball starting at 4R, passes the gap with action left, resting at 3R. After down it rolls to 3D. Finally, after left, it rolls to the cup at 3L but doesn't stop there due falling through the gap to 4L.
3. Three balls starting at 4U, 4U and 4U, under the action sequence right, left, up all end at 3U.
4. Three balls starting at 1R, 2U, 3D under the action up, end at 2U, 2U, and 2D.

Answer the following:

a) Describe the belief space of this problem with N=1. What does each belief state represent? How many unique belief states exist?

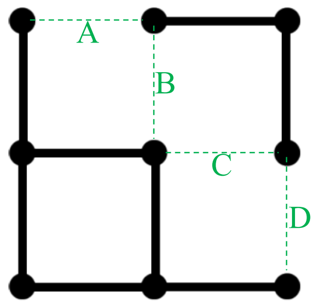
b) Starting from a complete lack of knowledge, sketch the four belief states that result after applying each action once (limit turtle art – simple drawings or lists of 2-digit poses).

c) How does the size of the belief space scale with N assuming we care only about the number of balls and not their unique identities)?

d) Describe a Conformant Plan leading every ball to the goal or argue why this cannot be done.

Question 3: Search and Game Playing [30]

You are playing the dots and boxes game on a 3x3 grid shown below. Each player must draw an edge connecting two dots, if it doesn't already exist. The player who draws the 4th line that completes a square receives a +1 score. The player with the largest score after none of the players can draw edges anymore wins the game.



It's the max player's turn.

a) Apply the Minimax algorithm to the above state, where the labeled and dotted lines are the four remaining options to play. Draw the corresponding search tree using the order A to D to organize your diagram.

b) Apply the alpha-beta pruning method using the same order of node expansion and show the alpha-beta values for all nodes. Is there any advantage to using alpha-beta pruning with this ordering? Can you find an ordering that leads to more pruning (or say why not)?