Artificial Intelligence

Name and ID: Viraj Amarasinghe(U14806439) / Jonathan Koch(U25318998)

Assignment 3

Due date: Mar. 5 at 11:59pm
You may work alone or in groups of two.

Problem 1 (2 points)
Read Chapters 4 and 5 of the textbook and answer the following questions.
(a) Explain the main differences between puzzle-solving search and game playing.

Puzzle-solving search and game-playing both focused on achieving a goal through a sequence of actions. However, they are not the same when we look at their objective, the given information, the environment, the complexity of the heuristics, the adversary, and the time constraints.

In puzzle-solving search, it uses a static environment or fixed set of constraints to solve a given problem. In-game playing focuses on winning the game by making better moves that would challenge their opponents, and players try to guess the best decisions. So, we can see that their objectives differ from each other.

Regarding the information, a puzzle-solving search gives all the information to solve the problem in the beginning but in the game-playing search, we do not get all the details upfront, and it involves uncertainty.

Another main difference between these two searches is the adversary. In puzzle solving, we do not have an adversarial nature, while game playing is involved an adversarial nature. Because the game players must play against one another, and they try to stop one another from winning.

Heuristic-wise, in the puzzle, solving it is possible to form an effective heuristic function that leads to the solutions. In the game, playing heuristics is more complex and sophisticated.

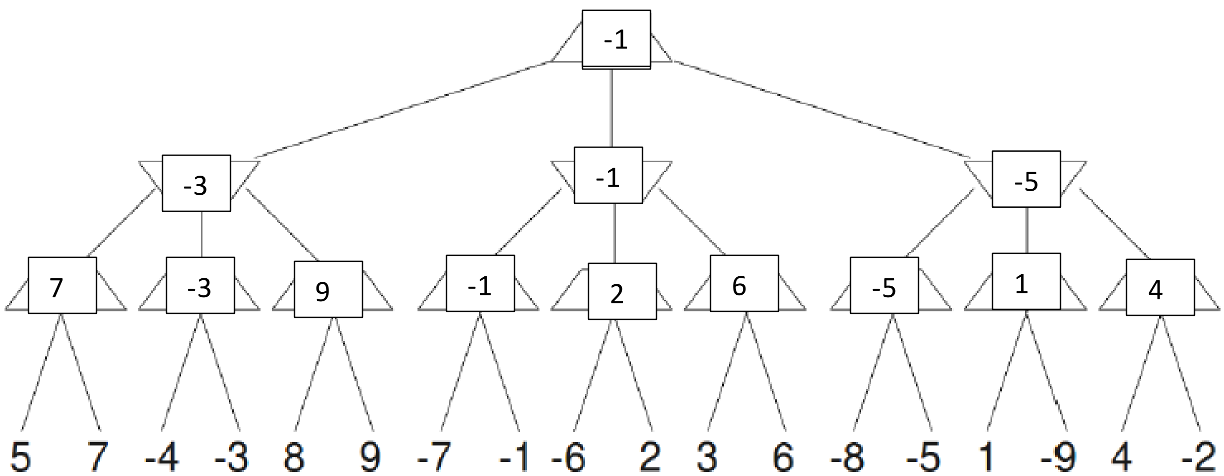
Puzzle-solving does not subject to time constraints, but game-playing searches do.

(b) Argue that game playing is inherently more difficult than solving one-player puzzles.

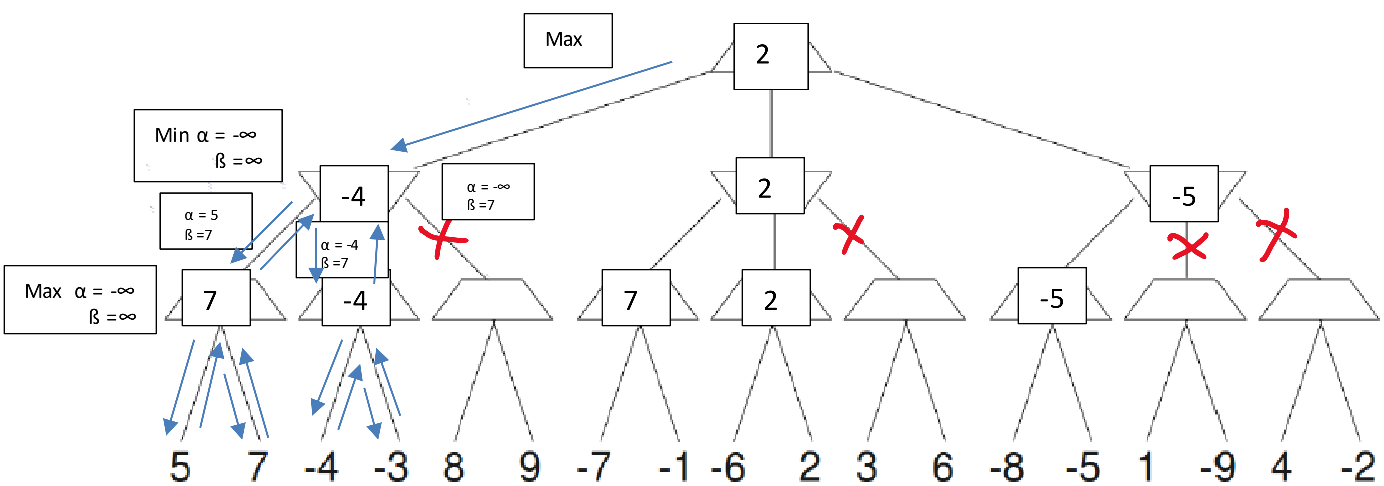
Game playing is definitely more difficult than a one-player puzzle. One of the major reasons is that game playing is involved two players and both players try to win over them, for that, they make their best moves and try to stop the other player from winning. When we have such a situation, we play against another same type of agent who observes the other player and makes decisions based on the opponent's moves, it becomes harder to achieve the objective.

So, in game playing, there are so many uncertainties and more complexity. But puzzle solving does not have such uncertainties and complexity. So, game playing is always more difficult than one-player puzzles.

Problem 2 (3 points)
Consider the zero-sum game tree shown below. Trapezoids that point up, such as at the root, represent choices for the player seeking to maximize; trapezoids that point down represent choices for the minimizer.
(a) Assuming both opponents act optimally, carry out the minimax search algorithm. Write the value of each node inside the corresponding trapezoid and highlight the action the maximizer would take in the tree.



(b) Now reconsider the same game tree but use alpha-beta pruning. Expand successors from left to right. Record the alpha-beta pair that is passed down that edge (through a call to MIN-VALUE or MAX-VALUE). Circle all leaf nodes that are visited. Put an `X' through edges that are pruned off.



(c) True / False. Minimax and alpha-beta pruning are guaranteed to find the same value of the top node.

TRUE

Problem 3 (15 points)
The *hexagon game* involves two players, who gradually construct a six-vertex undirected graph with solid and dashed edges. Player 1 adds solid edges, whereas Player 2 uses dashes.

The players begin with a six-vertex graph that has no edges (Figure 1), and add new edges, one by one; Player 1 makes the first move. At each move, a player has to add a new edge between two vertices that are not connected by any old edge. Figure 2 is an example of a mid-game position, where Player 1 has to make the next move.

If Player 1 constructs a solid-line triangle, he loses the game; similarly, a dashed triangle means a loss of Player 2. For example, if the game ends as shown in Figure 3, then Player 2 has lost since he has constructed the dashed triangle "3-5-6."

Implement a program for playing the hexagon game. Your program should prompt the user to enter a player number (1 or 2), and then act as the specified player. For example, if the user enters "1", the program should make the first move.

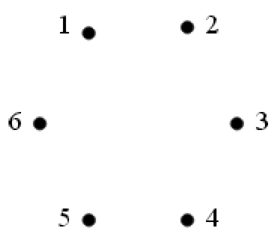


Figure 1: Initial state.

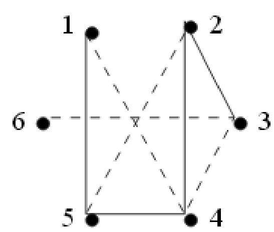


Figure 2: Mid-game position; Player 1 makes the next move.

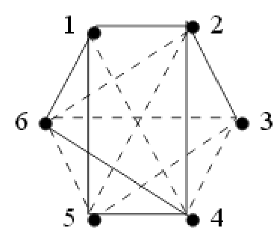


Figure 3: Player 1 has won.

Submitting your assignment

- Submission via Canvas Assignment.
  - It is your responsibility to submit these assignments in a timely fashion.
- All files should be zipped together.
- Name of your file must include your last name and ID number
- There should be a readme file explaining in detail the exact steps to be taken to compile and execute the code files and the title page
- In case of any code errors, partial credit may be offered based on the code and documentation.
- A report that presents the performance evaluation of your solution.

Late Submission Policy

- Late work will be not accepted.

Rubric for Assignment 3

Problem 1 (2 points)

a. [1/]
b. [1/]

Problem 2 (3 points)

a. [1/]
b. [2/]

Problem 3 (15 points)

- Compile, run and test (15 points)

  [5/] code compile
  [5/] Run the program
  [3/] Test cases
  [2/] Report