**Artificial Intelligence**

Name and ID Georgy U56820310

Jinhao U94977930

**Assignment 3**

**Due date: March 26 at 11:59pm**

You may work alone or in groups of two.

**Problem 1** (4 points)

Read Chapters 4 and 5 of the textbook and answer the following questions.

1. Explain the main differences between puzzle-solving search and game playing.

ANS:

With a puzzle-solving search, its single player so the agent knows the rules, knows what

Actions to do, know when you win. A tree simply shows the options of what you can do.

Game playing requires knowing what to do base on the situation one finds oneself in.

Puzzles are made to be solved.

Games are different. Games pack surprises, even for experienced players. When you play a game, you’re not just executing an algorithm. You’re thinking, strategizing, discovering.

Games require the ability to make some decision even when calculation the optimal decision is infeasible.

1. Argue that game playing is inherently more difficult than solving one-player puzzles.

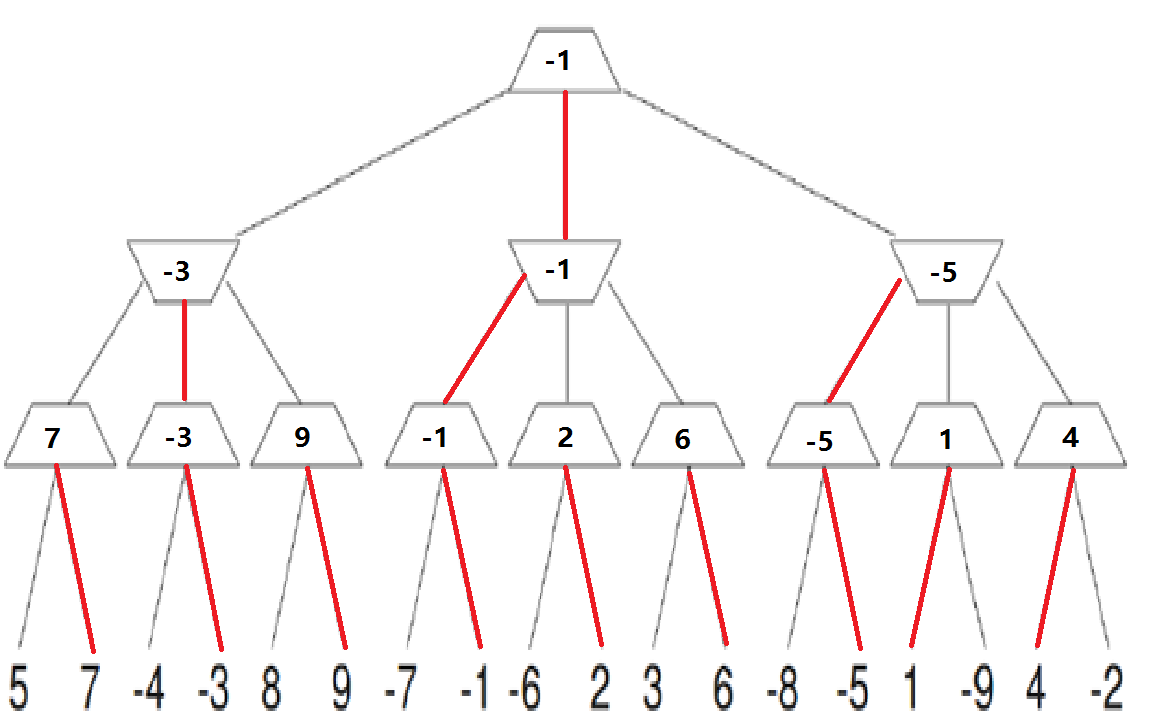
ANS:

Game playing is more difficult than solving a one-player puzzle because the outcome isn't affected solely by your decisions; another player changes the current situation as well. Game playing requires not just a sequence of actions like puzzle-solving, but a contingency plan knowing what to do base on the state one finds oneself in.

**Problem 2 (6 points)**

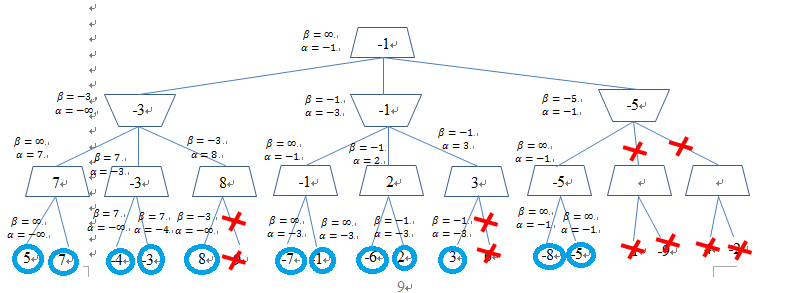
Consider the zero-sum game tree shown below. Trapezoids that point up, such as at the root, represent choices for the player seeking to maximize (MAX); trapezoids that point down represent choices for the minimizer (MIN).

(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.

****

**Report:**

The idea of our project that we define a hexagon of size 6 by 6 which is represented as 2D int array. We have implemented function Lost that checks if the player has connected a triangle. Than we implemented player turn function that validates the turn. Then there is a function for a PC turn that checks all possibilities and choose one that does not lead to lose.