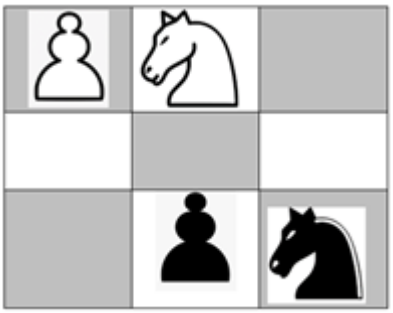
Artificial and Computational Intelligence
Assignment 2

Problem statement 1 : Gaming

Simulate the working of chess with the below sample smaller game board. A player with any piece/coin standing where its opponent sacrifices all the piece/coin is a win for the first player.

Soldier/Pawn (S) can move only one straight step ahead. Only if opponent coins are available, it can move in diagonally one step forward to attack the opponent coin.
Horses/Knight(H) can move either forward/backward in L–shaped strides covering exactly 3 tiles. It is the only piece that can jump over other coins



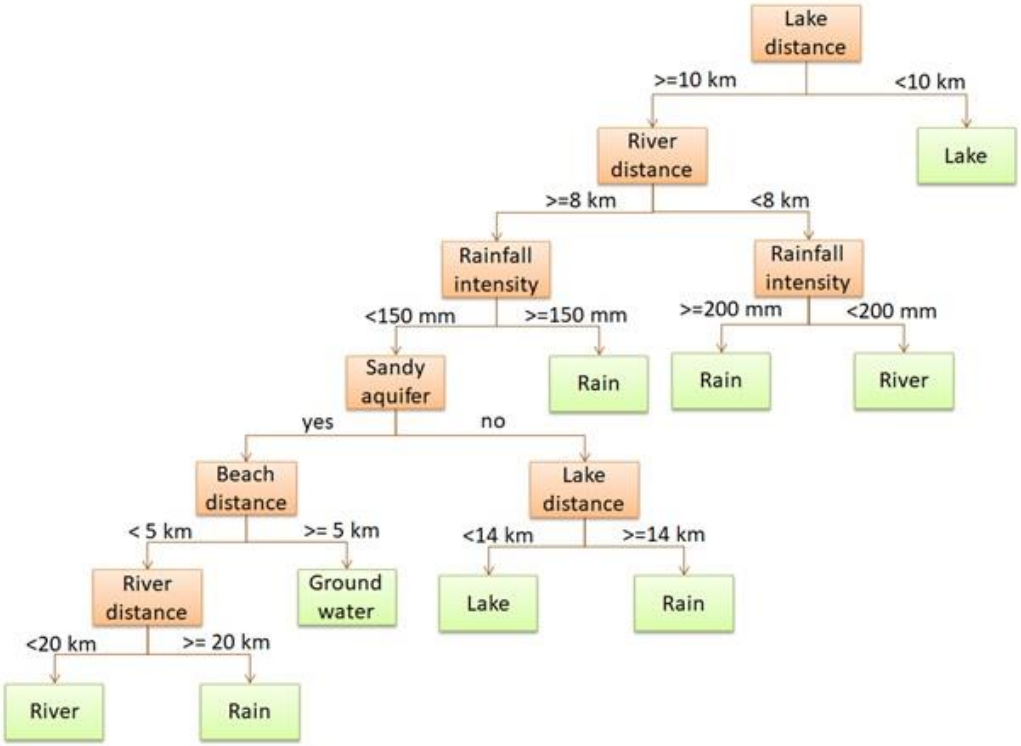
a. You are free to choose your own static evaluation function. Justify your choice of static evaluation value design and explain with a sample game state. Do not use any machine learning model for the evaluation function.

b. Similar to the virtual lab example, one of the players must be a human ie., it must get dynamic inputs from us. The other player must be simulated using the program.

c. Implement Python code for the design under part a, using Minimax Algorithm

Problem statement 2: Logic

The figure below is a Decision Tree created to predict which water resource a location is most suitable to utilize. For example, if there is a case where a location has features: 120 mm/month rainfall, sandy aquifer, 10 km away from the perennial river, 20 km away from the lake, and 2 km away from the beach, could you decide which water resource suitable for the community to take from, rainfall, river water, lake, or groundwater? Use the below decision tree and create Prolog rules to predict which water source is best for the community under the given conditions. Take the attribute values from user by giving suitable user prompts and predict the water source.



Reference: https://www.analyticsvidhya.com/blog/2021/04/distinguish-between-tree-based-machine-learning-algorithms/

Evaluations will be based on the following.

1. Use Min-Max algorithm and implement the game in PYTHON (35% marks)
2. Derive the rules from the given decision tree and code as Prolog rules. (35% marks)
3. Interactive implementation. Dynamic inputs-based run of the game with step wise board display and error free game ending. (15% marks)
4. Interactive implementation. Dynamic inputs-based run of the logic expert system with step wise options display and error free recommendation & ending. (15% marks)

Important Note:

- You are provided with the python notebook template which stipulates the structure of code and documentation. Use well intended python code.
- Use a separate MS word document for explaining the theory part. Do not include the theory part in the Python notebook except Python comments.
- The implementation code must be completely original and executable.
- Please keep your work (code, documentation) confidential. If your code is found to be plagiarized, you will be penalized severely. Parties involved in the copy will be considered equal partners and will be penalized severely.