Samples for Quiz 2

Task 1 [Included for Mon/Wed sections. Already covered in Quiz 1 for Tue/Thu sections]

Consider a search space, where each state can be red, green, blue, yellow, or black. Multiple states may have the same color. The goal is to reach any black state. Here are some rules on the successors of different states, based on their color (these successor functions are unidirectional):

- Red states can only have green or blue children.
- Blue states can only have red or black children.
- Green states can only have blue or yellow children.
- Yellow states can only have yellow or red children.
- Black states can only have green or black children.

Define a maximally admissible heuristic that assigns a value to each state based only on the color of that state

Assume that any move from one state to another has cost 1.

Task 2

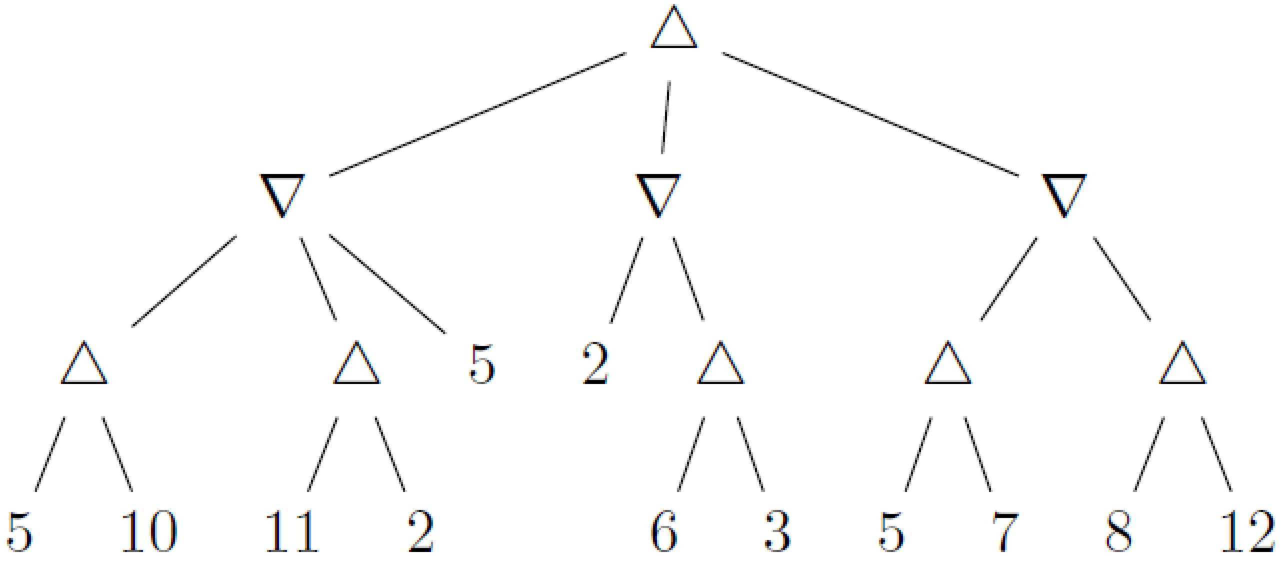

Figure 1: A game search tree.

a. In the game search tree of Figure 1, if we use Minmax algorithm to find the strategy to use, indicate the minmax values are for all the nodes. Also incidcate which action the Minmax algorithm will pick to exectute.

b. In the game search tree of Figure 1, indicate what nodes will be pruned using alpha-beta search, and what the minmax values are for the rest of the nodes. Assume that alpha-beta search expands nodes in a left-to-right order. Finally incidcate which action the Minmax algorithm will pick to exectute. Is this answer different from part a?

c. This question is also on the game search tree of Figure 1. Suppose we are given some additional knowledge about the game: the maximum utility value is 12 i.e., it is not mathematically possible for the MAX player to get an outcome greater than 12 and the minimum utility value is 2 i.e., it is not mathematically possible for the MIN player to get an outcome lower than 2. How can this knowledge be used to further improve the efficiency of alpha-beta search? Indicate the nodes that will be pruned using this improvement. Again, assume that alpha-beta search expands nodes in a left-to-right order.

Task 3

Suppose that you want to implement an algorithm that will compete on a two-player deterministic game of perfect information. Your opponent is a supercomputer called DeepGreen. DeepGreen does not use Minimax. You are given a library function DeepGreenMove(S), that takes any state S as an argument, and returns the move that DeepGreen will choose for that state S (more precisely, DeepGreenMove (S) returns the state resulting from the opponent's move).

Write an algorithm in pseudocode (following the style of the Minimax pseudocode) that will always make an optimal decision given the knowledge we have about DeepGreen. You are free to use the library function DeepGreenMove(S) in your pseudocode. What advantage would this algorithm have over Minimax? (if none, Justify).

Task 4

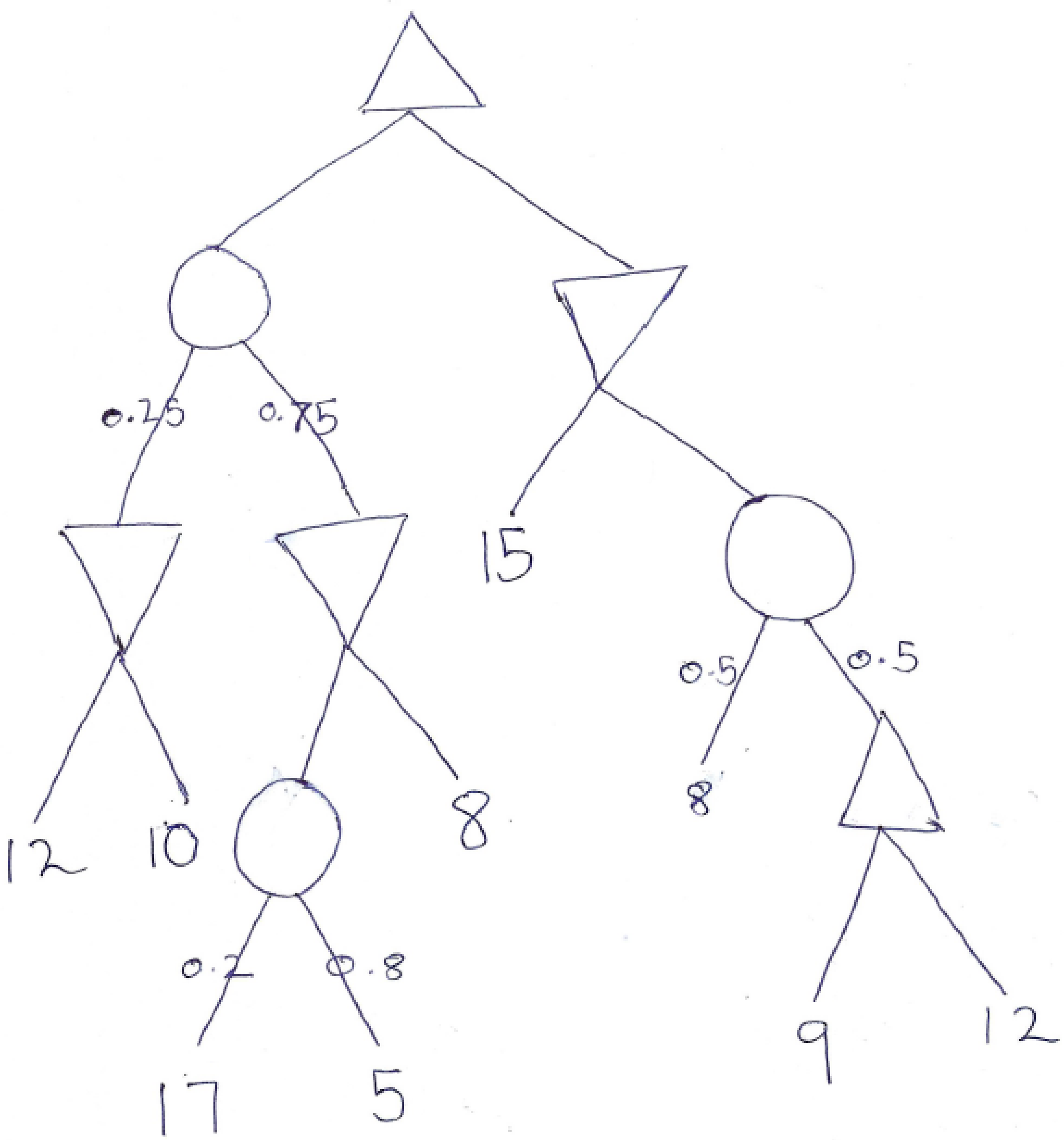

Figure 2: An Expectiminmax tree.

Find the value of every non-terminal node in the expectiminmax tree given above. Also indicate which action will be performed by the algoirithm. What is lowest and highest possible outcome of a single game if the minmax strategy is followed against an optimal opponent?

Task 5

Max: [4308: 35 Points (+5 Pts EC), 5360: 35 Points (+5 Pts EC)]

Consider the following map.

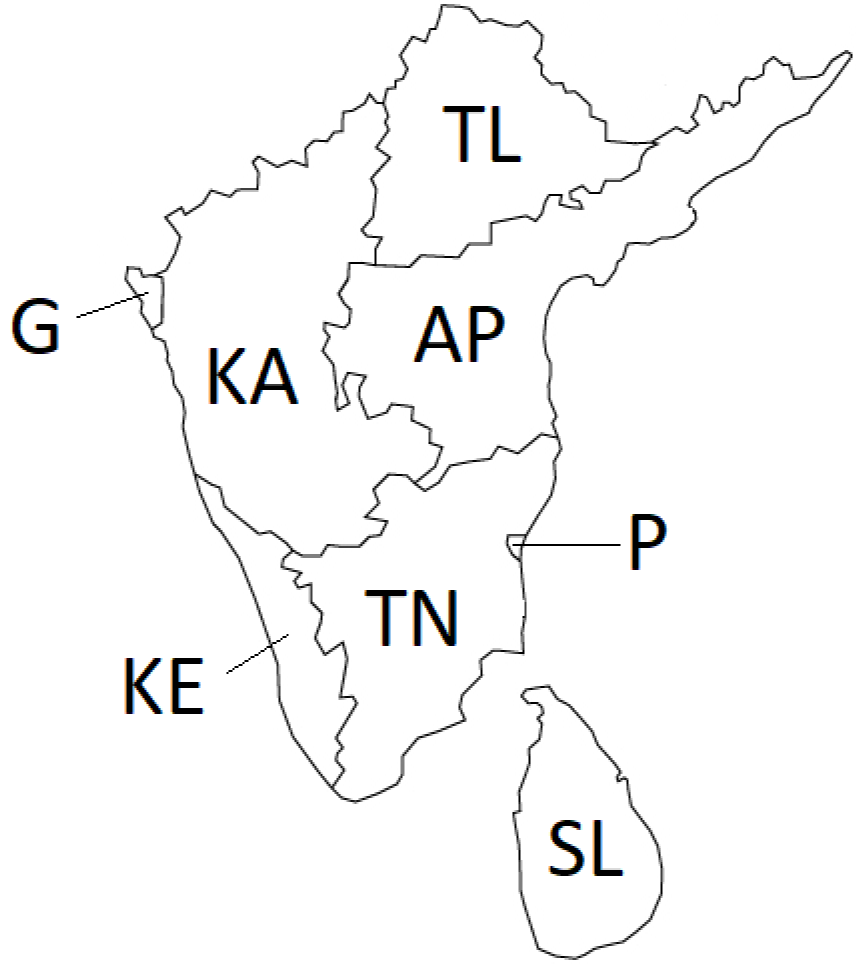

Figure 3: Outline of a Map

The problem is to color the sections such that no two sections sharing a border have the same color. You are allowed to use the colors (Red, Green, Blue).

Part a: Draw the Constraint Graph for this problem.

Part b: Assuming you are using Backtracking search to solve this problem and that you are using MRV with Degree heuristic to select the variable, Which variable will be selected at each level of the search tree [You do not need to draw the tree. Just let me know which variable will be selected and why (MRV and degree values)]. Note: Multiple possible correct answers. You only have to give one.

Part c: Assume you assign the color 'Red' to the first variable selected in part b. Show the steps involved in checking the remaining legal values for all other variables using Arc Consistency.

Part d: Can you use structure of the problem to make solving it more efficient?

Part e: Give one valid solution to this problem. (You just have to give the solution. No need to give all the steps)