Homework 02

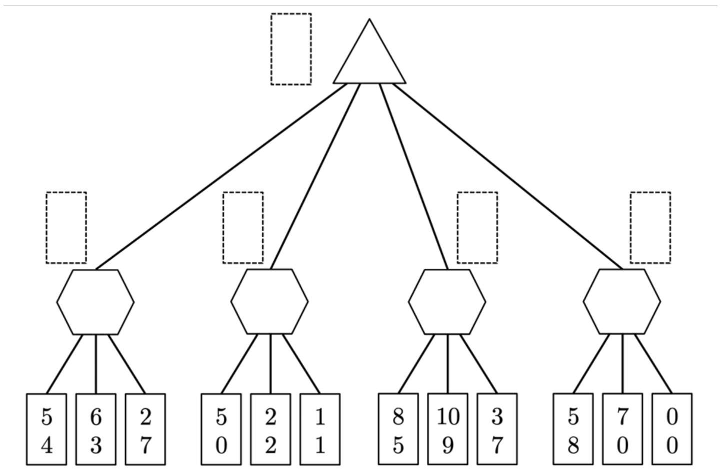
Problem 1. (3.0pts) [Local search] Answer the following questions about local search algorithms

a) Which places in the state space make regular hill climbing fail to the optimal solution?

b) Are variations of hill climbing (such as Random-restart hill-climbing, local beam search, etc.) guarantee complete and/or optimal? Briefly explain.

c) It is possible to get stuck in a local maximum in simulated annealing. Is it TRUE or FALSE? Briefly explain.

Problem 2. (3.0pts) [Adversarial Search] You are an excellent programmer and you program a robot to play a scoring game with you. You and your robot both have your own score. You are trying to maximize your score and do not care about your robot' score. On the other hand, your robot tries to minimize the absolute difference between the two scores. In case of a tie, the robot prefers a lower score. For example, the robot prefers (4,3) to (5,3) since the abs(4-3) is less than abs(5-3); it prefers (2,3) to (4,3) since a tie occurs and (2, 3) is lower than (4,3); and it prefers (7, 7) to (9, 9) because a tie occurs and (7, 7) is lower than (9, 9).



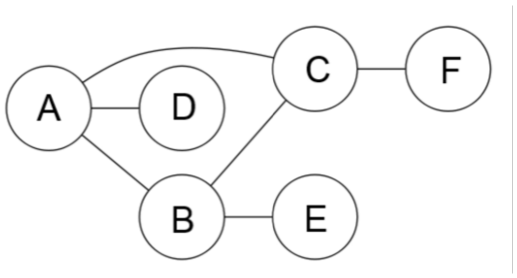
The above tree shows the game tree. The root node is your max node and the below nodes are the robot's nodes for four different actions. The leaf nodes are the scores with your score on top and the robot' scores are on the bottom.

a) Fill in the dashed rectangles with the pair of scores preferred by each node of the game tree.

b) You can save computation time by using pruning in your game tree search. Is it possible to prune any branch on the tree? If yes, please mark the branches with an 'X'.

c) You now have access to an oracle that tells you the order of branches to explore that maximizes pruning. On the copy of the game tree above, put an 'X' on line of branches that do not need to be explored given this new information from the oracle.

Problem 3. (2.0pts) [Constraint Satisfaction Problem]



The graph above is a constraint graph for a CSP that only has binary constraints. Each node represents a variable. At the beginning, the variables have not been assigned any value.

Please answer the following questions, given that assigning values to the variables results in their domain being changed.

a) A value is assigned to A. Which domains might be changed as a result of running forward checking for A?

b) A value is assigned to A, and then forward checking is run for A. Then a value is assigned to B. Which domains might be changed as a result of running forward checking for B?

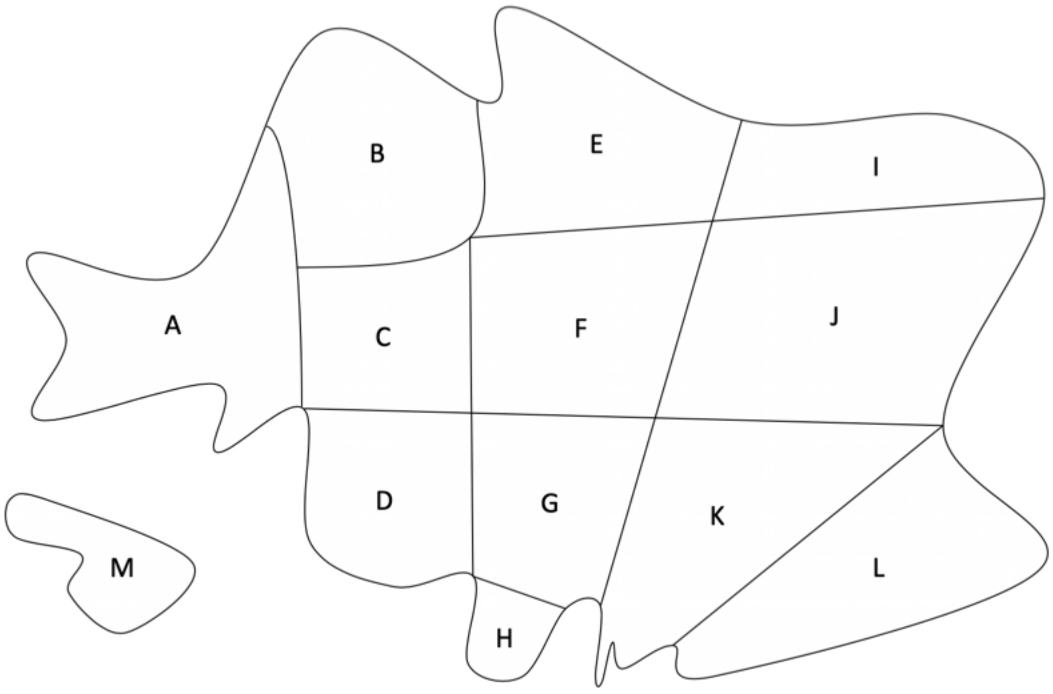
c) A value is assigned to A. Which domains might be changed as a result of enforcing arc consistency after this assignment?

d) A value is assigned to A, and then arc consistency is enforced. Then a value is assigned to B. Which domains might be changed as a result of enforcing arc consistency after the assignment to B?

Problem 4. (2.0pts) [Map coloring – Constraint Satisfaction Problem]

In the given map below, there are 14 regions corresponding to 14 capital letters (from 'A' to 'M').

a) Please find the minimum number of colors needed to color the regions with the constraint that no two adjacent regions have the same color. You just need to state the minimum number and give one sample of the colors assigned to each of the regions that satisfy the constraint.



b) Assuming that there are three colors: red, blue, and green. Initially, we can give every region one of the colors. It means that the color domains of each of the regions are {red, blue, green}. Then, we assign the region F to have green color. What is the result of the Forward Checking algorithm?