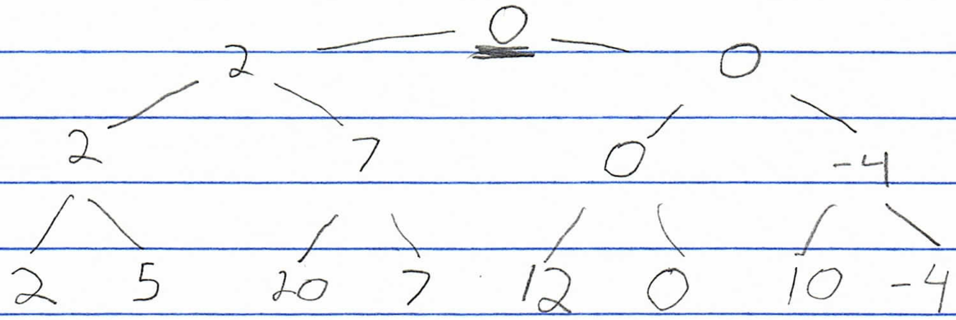
Gabriel Shultz

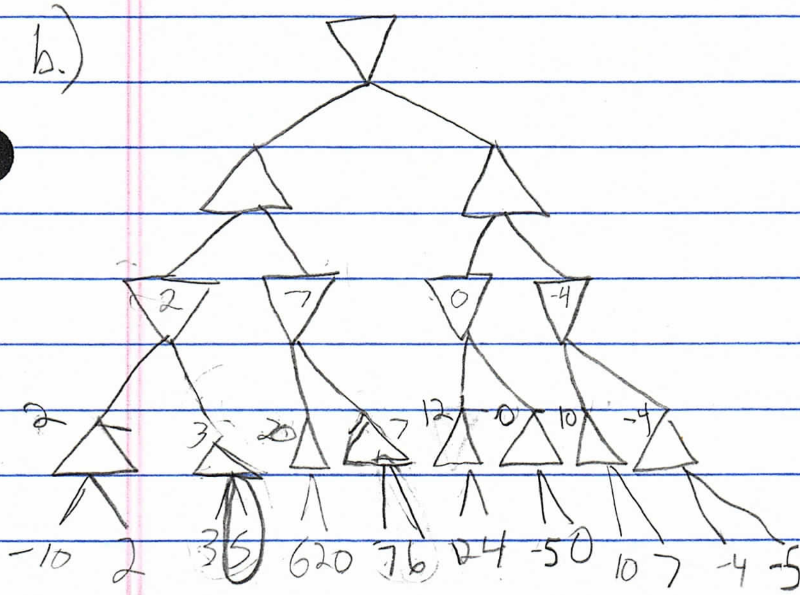
Homework 3 AI

1a)



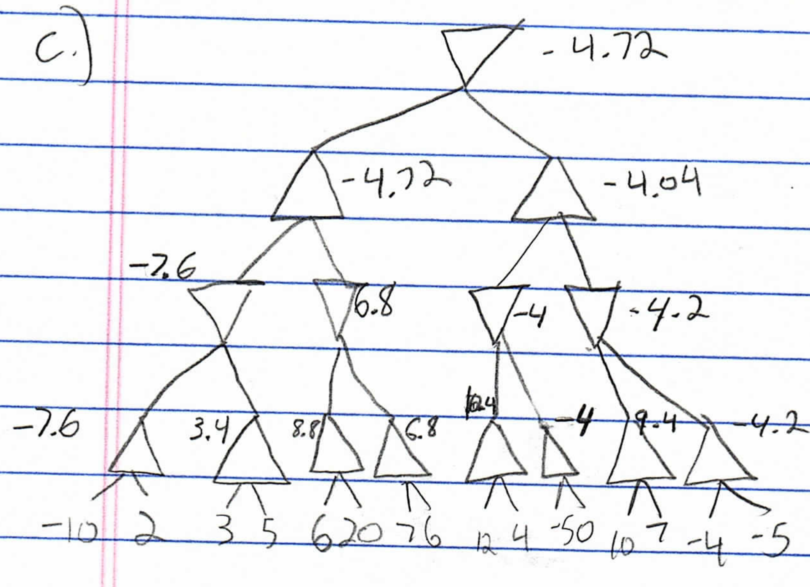
a.) The value state at the root will be 0. Minimax search would be a suitable algorithm.

b.)



Yes we can use alpha beta pruning, but only the branch with the leaf "5" can be pruned/not expanded.

c.)



d.) No pruning can happen on a stochastic tree because each leaf can skew the average.

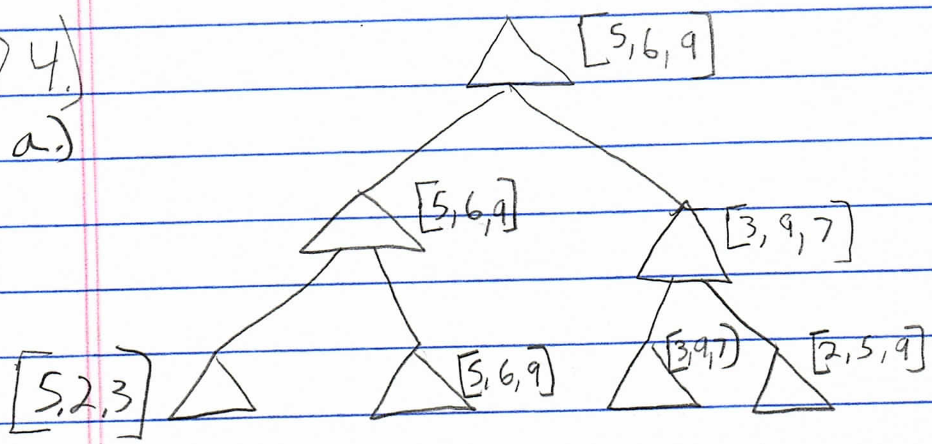
Q3.)

$$\# \text{ of leaves to explore} = (b)^{d-1} + (b-1)$$

In a perfect situation, all b nodes on the left most parent must be explored. Then only 1 child on each other second to lowest level parent must be checked. There are $(b)^{d-1}$ parent nodes on the second to last level. Only 1 of their children need to be explored. The b-1 term corrects for the leftmost parent who explores all their children.

Q 4.)

a.)



b.) An initial thought I had was when computing the decisions for the C agent, assume the tree is taller and C will actually get another turn. Well, if a previous C agent made a decision where A and B were higher than the tuple you were currently looking at, then you know that this leaf will never occur, so you dont have to bother looking at the 3rd # in the tuple. You still have to look at the tuple though. So I am unsure if this is "Pruning." But you could save a couple of comparisons. You would still have to look at all tuples in the leaves, but you may not have to look at that particular agent's element in the tuple.

Q5

a.) False, alpha-beta pruning still returns the optimal strategy same as minimax. Both rely on all agents being rational.

b.) False, the ordering of the children directly impacts the number of child nodes expanded. If the current agent is a min and it knows a previous instance of itself chose a 3 on the same level, and it sees a 2 in its child node, it can prune the rest of its children because it knows that max will not pick this path. If the value 2 was in a later child, it may not have been able to make the decision to prune as early.

c.) True, alpha beta pruning assumes that your opponent will make the best decisions for themself. However, if they don't, the utility may not be as good as it could be, but will still be the same or better than playing with an optimal MIN, never lower.

d.) False, with stochastic decisions every leaf needs to be explored as the weighted average could be drastically changed unless there is a bound on the leaves values.

Q6.)

a.) Never

b.) Never

c.) Sometimes

d.) Sometimes

Q7.) Variables: {Mom, Dad, Baby, Student, Teacher, Guide}

Domain: {1, 2, 3, 4, 5, 6}

between parents

student and teacher next to each other.

Constraints {Mom < Baby < Dad $\vee$ Dad < baby < mom,
student = teacher + 1 $\vee$ student = teacher − 1,
Guide = 1 $\vee$ Guide = 6}

Q8.) Variables X: {A, B, C, D, E}

Using other time slot notation

Domains: R: $\{R_1, R_2\}$, T: {1, 2, 3, 4}

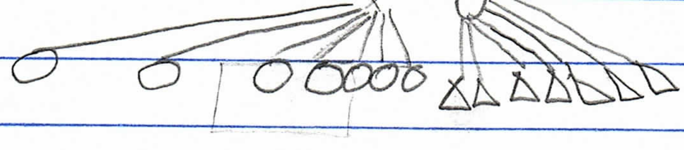
Constraints $\{ A_T <= 2 ;\ B_T = 1 ;\ D_T \geq 3 ;$
$A_T < C_T ;\ \{A, B, C\}_R \neq R_1 ;\ \{D, E\}_R \neq R_2 ;$
$a_{TR} \neq b_{TR}$ for $a, b \in X \}$

Q2

a.) There are 9 possible positions on a tic-tac-toe board. After each player plays once there are only 7 states left. In other words the game has $2\binom{9}{2}$ different states (the factor of 2 accounts for the switching of X's and O's. At least this would be the case if each player had a unique symbol. Instead, a coin flip is used. This scenario leads to another factor of 2 to account for the cases where there are 2 O's or 2 X's instead of one or another. $4\binom{9}{2} = 144$

So, at the end of each players first turn, the game has 144 possible states.

Branching factor decreases after each turn though. For instance, the first agent has 9 placement options after the coin toss. And the second has 8.
At the start of the 3rd move, the tree will have



where O, $\Delta \in$ 144 states

b.) Minimax, even though there is probability involved, each agent plays optimally and can calculate its deterministic best option. c.) Yes the win will involve chance but there is an optimal