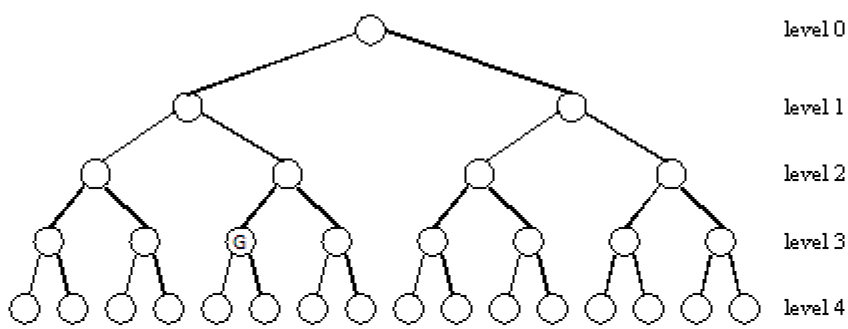
Due Tuesday night April 30 via Gradescope at 11:59 PM. A maximum of one late day may be used on this assignment, with the usual 10 percent reduction in score.

Do the following exercises. These are intended to take 15-25 minutes each if you know how to do them. Each is worth 25 points.

---

1 Efficiency of Search Algorithms

Suppose a binary tree-searching algorithm needs to search to a maximum depth of 4. However, it could find a goal node and stop there, at any level d of the tree, $0 \leq d \leq 4$.



(a) (3 points) How many nodes in the tree above must be visited by each of the following algorithms, to find the goal node marked G? (Assume children are handled in left-to-right order.)

DFS: 11 BFS: 10 IDDFS: 18

(b) (6 points) Now for a general tree having N vertices, and any branching factor, including 1, or even irregular (varying numbers of children, but obviously always non-negative numbers), Give Big-Theta characterizations of the worst-case running times:

DFS: $\Theta(N)$ | BFS: $\Theta(N)$ | IDDFS: $\Theta(N^2)$; worst-case is when $b = 1$, i.e., no branching

(c) (6 points) Using Big-Theta notation again, describe the amount of memory required by each algorithm for the general tree of N vertices.

DFS: $\Theta(N)$ | BFS: $\Theta(N)$ | IDDFS: $\Theta(N)$

(d) (3 points) Suppose we now want to use graph-search versions of these algorithms. What is an important benefit of BFS and IDDFS that DFS does not have?

They are guaranteed to have found a shortest path from start to a goal when they first reach a goal (assuming predecessor links are stored appropriately.)

(e) (3 points) What impact does a heuristic function have on the run time of the A* algorithm? Assume that the heuristic is admissible.

A good heuristic may bias the otherwise blind search to prioritize exploration towards the goal, and therefore process fewer nodes, while still determining a lowest-cost path when first arriving at a goal node.

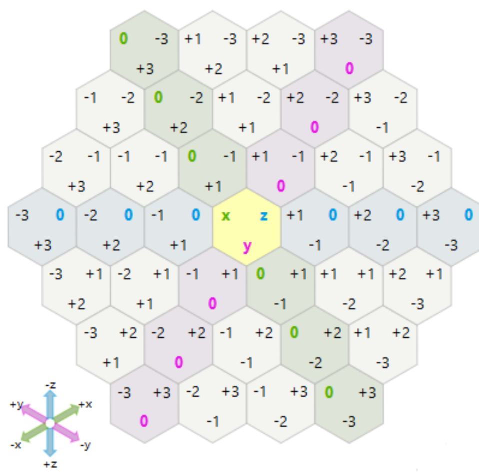
(f) (4 points) What does an admissible heuristic do for us in the best and worst cases?

Best case: avoids expanding any nodes off the optimal path, and doesn't cost much to compute the heuristic; worst-case: not only fails to avoid expanding any nodes expanded by Uniform-Cost Search, but costs extra time to compute the heuristic.

2 Heuristic Search

Imagine that you are part of a game-design team creating a game that takes place in a hexagonal grid world. Your job is to evaluate some heuristics proposed to help the monsters of the game do path planning.

The figure below shows a general hexagonal grid in which each cell is identifiable by a triple of coordinates: x (which tells how far it is from the origin to the right (or left), y (which tells how far it is from the origin on an axis that makes an angle of $\pi/3$ with the x axis, and z (which tells how far from the origin it is on an axis that makes and angle of $2\pi/3$ with the x axis). Note that there is (intentionally) some redundancy here. In particular, $x + y + z = 0$.



We define the distance between two cells as follows:

$$d((x_1, y_1, z_1), (x_2, y_2, z_2)) = |x_1 - x_2| + |y_1 - y_2| + |z_1 - z_2|$$

The monsters in the game need to perform path planning for two reasons. One is to escape from fighting with the protagonist. The other is to move towards the protagonist to make an attack. (Although the monsters should not be too intelligent, we will assume that they do need to know what the lowest-cost path is between their starting states and destination states.)

Suppose the operators are the following, in the sequence given. (This order has significance during the various searches.)

ϕ_0: $x \leftarrow x+1$ and $y \leftarrow y-1$.

ϕ_1: $x \leftarrow x+1$ and $z \leftarrow z-1$.

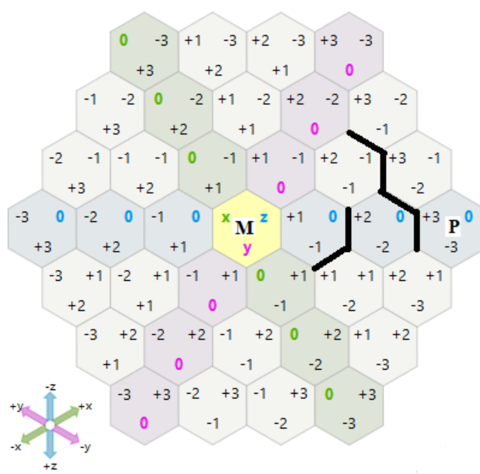
ϕ_2: $y \leftarrow y+1$ and $z \leftarrow z-1$.

ϕ_3: $y \leftarrow y+1$ and $x \leftarrow x-1$.

ϕ_4: $z \leftarrow z+1$ and $x \leftarrow x-1$.

ϕ_5: $z \leftarrow z+1$ and $y \leftarrow y-1$.

(a) (8 points) Consider the problem of finding a lowest-cost path from M to P in the example game situation below. The thick black walls in the diagram represent barriers through which the monsters must not pass. (Therefore, the precondition for operator ϕ_0 is not satisfied at state $(+1, -1, 0)$, and neither is the precondition for operator ϕ_5, so that this state has only 4 successors.)



What is the cost of the shortest path? (i.e., total distance for all its edges)
8

What is the number of states that would be expanded by BFS to find a lowest-cost path from M to P?
36

(b) (4 points) Consider the heuristic $h_x((x, y, z)) = |x - x_g|$, where x_g is the x coordinate of the goal cell.

Determine whether or not h_x is admissible and explain why it is or why it is not.
h_x is admissible because it always underestimates the true distance to the goal. The distance between a cell and the goal cell is $|x - x_g| + |y - y_g| + |z - z_g| \geq |x - x_g| = h_x$

(c) (4 points) Determine how many states would be expanded by A* using h_x to find a shortest path from M to P.
23

(d) (3 points) Explain how the use of the heuristic is affecting the search (compared with BFS).
The heuristic influences the search by expanding states that are closer to the goal based on the x component distance it is away from the current cell. As a result, the search considers less states that are on the left side of the grid world and instead prioritizes expanding nodes on the right side of the grid first because these have a lesser x component distance to the goal.

(e) (3 points) Propose another heuristic that you believe will outperform h_x. (Do not propose the exact distance defined above or function that includes it.) Give a formula to define your function. Also give an argument for or against its admissibility. Why do you believe it will outperform h_x?
Many possible two-component heuristics possible; example $h_{xy} = |x - x_g| + |y - y_g|$

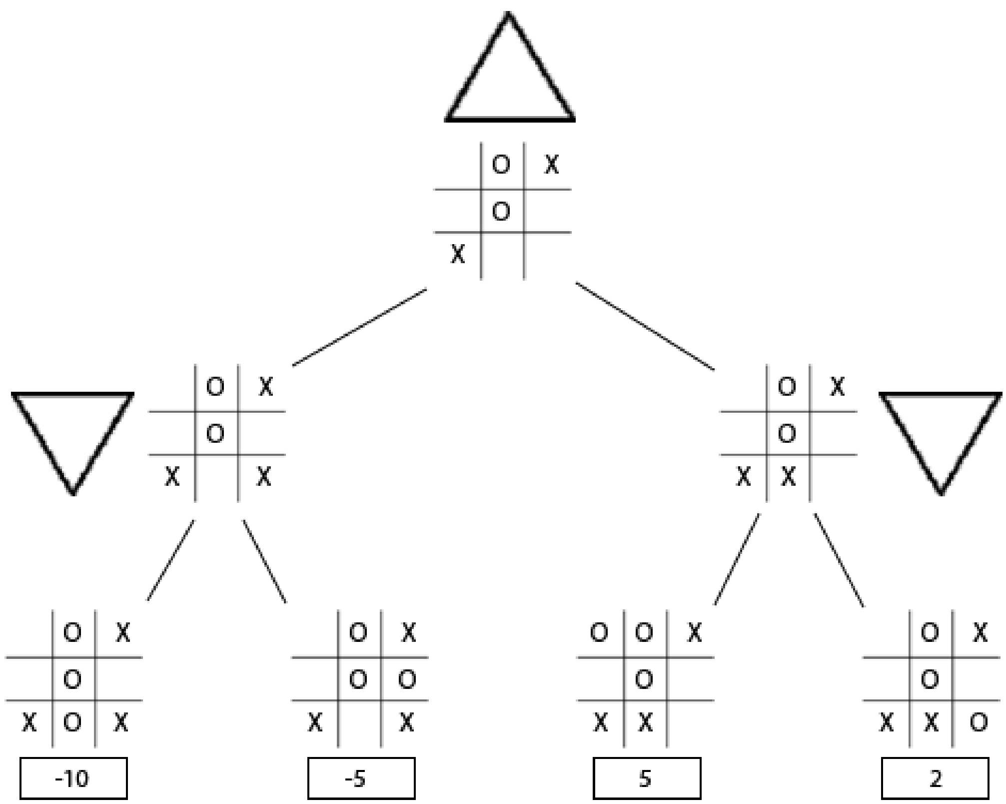
h_{xy} is admissible because it always underestimates the true distance to the goal; distance between a cell and the goal cell is $|x-x_g|+|y-y_g|+|z-z_g| \geq |x-x_g|+|y-y_g| = h_{xy}$

h_{xy} considers not only the x component distance between the current cell and the goal but also the y component distance which makes it a more accurate approximation of the true distance to the goal.

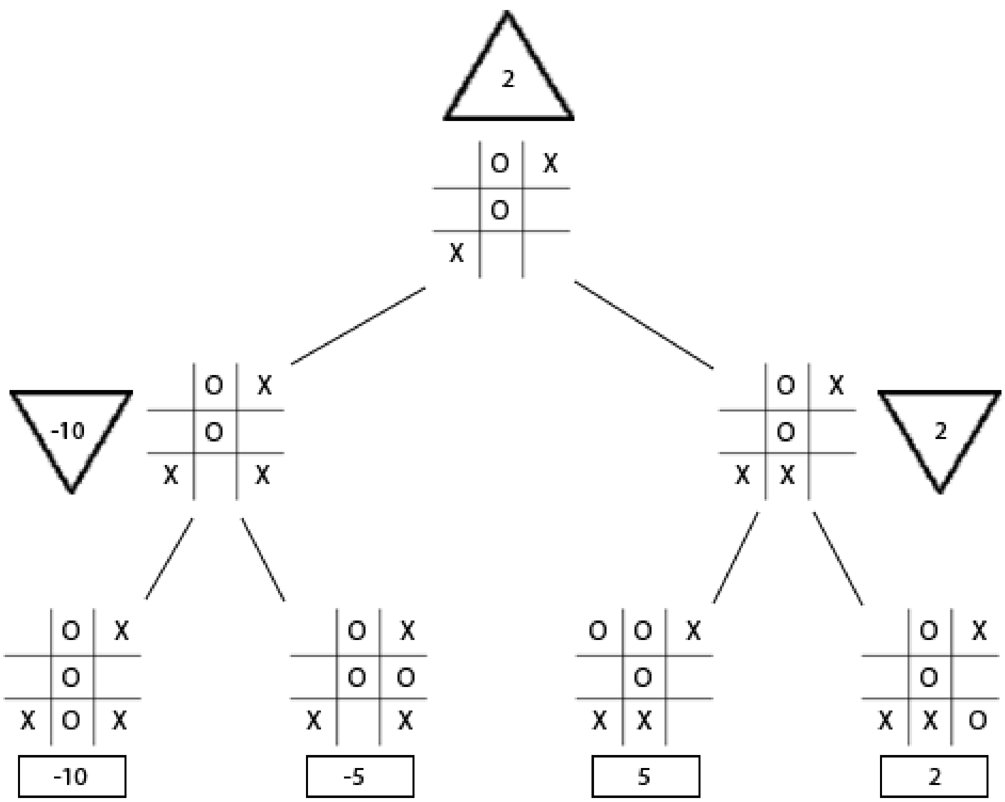
(f) (3 points) How many states will be expanded by A* using your heuristic to find a shortest path from M to P? $h_{xy} = 11$; $h_{xz} = 14$; $h_{yz} = 14$

3 Minimax Search

(a) **Basic minimax with static evaluation.** (9 points) Suppose you are in the middle of a game of Tic-Tac-Toe. You are X and trying to maximize the state value. Your opponent, O, is trying to minimize the state value. In the middle of the game you decide that you want to place X somewhere in the bottom row. However, you are unsure which spot to place X in. Complete the minimax tree below (i.e. fill in the triangles).



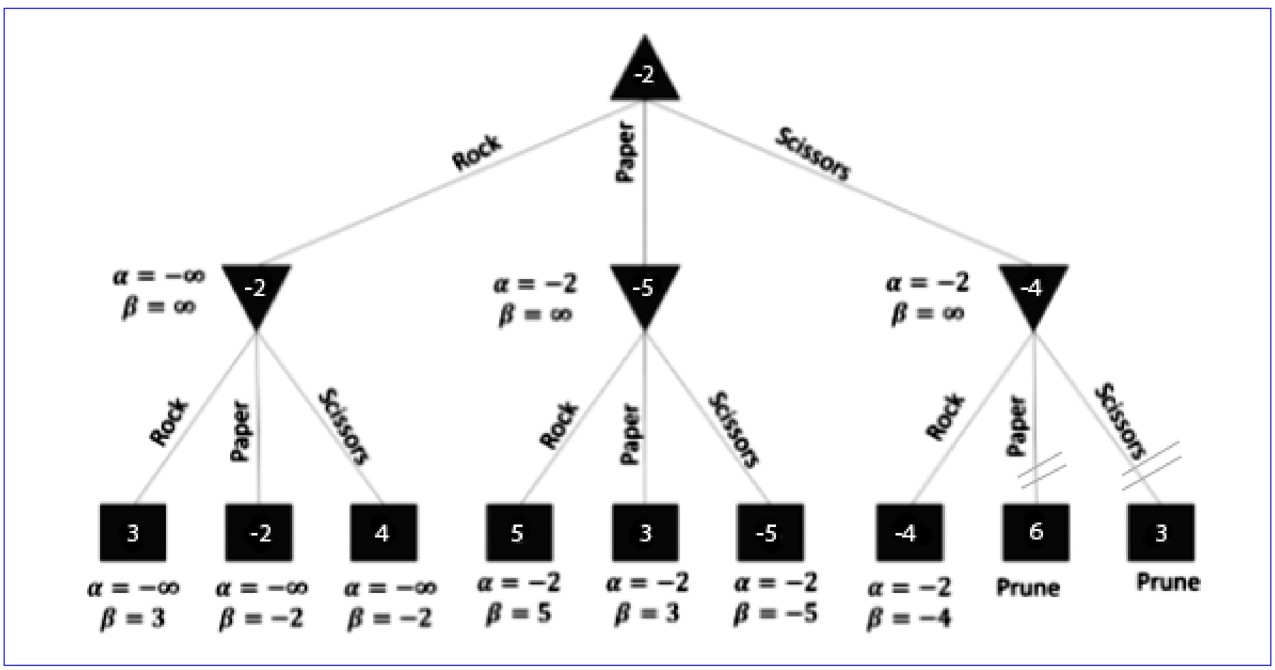
Solution:



(b) **Alpha beta pruning** (16 points) You now challenge a new friend to a game of Rock-Paper-Scissors. Once again you are trying to maximize the score while your opponent is trying to minimize it. The score corresponding to each case is given as follows:

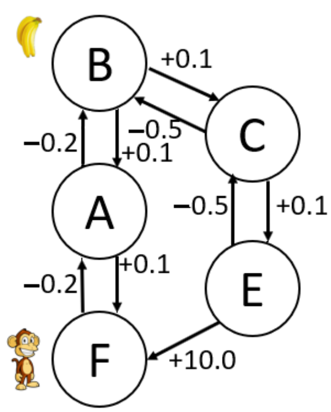
You	**Opponent**	**Score**
Rock	Rock	3
	Paper	-2
	Scissors	4
Paper	Rock	5
	Paper	3
	Scissors	-5
Scissors	Rock	-4
	Paper	6
	Scissors	3

In a modified version, the players take turns to show their moves i.e., you first choose among {Rock, Paper, Scissors}, and then the opponent chooses among {Rock, Paper, Scissors}. Draw the simplest search tree with the scores provided in the table above. Include the final alpha and beta values at each step and mark the pruned nodes with a double slash (//).



4 Markov Decision Processes

Consider a situation in which a robot monkey is in a room in which some bananas are hanging from the ceiling. The monkey can move its left arm to any one of 3 different places, and it might be holding or not holding bananas in either of two states. The monkey starts out not holding bananas with left hand low (state F). It can try to raise its hand, and so it can transition to state A. However, this monkey has a somewhat unreliable body, and most actions are only effective with probability 0.8. The action of raising the hand from state F, however, is the one completely reliable and deterministic action, so the next state after that action is guaranteed to be state A. If the monkey again tries to raise his hand from state A, there is a 0.8 probability of arriving in state B, which is roughly where the bananas are. There is a probability of 0.2 that the action will cause the monkeys hand to move back down to state F. From state B, the monkey has a choice of action: (T) take the bananas and go down to state C, or (L) leave the bananas and go down to state A. (These again are nondeterministic.) From state C, the monkey can go to state E or back up to state B. From state E, the monkey can finally get the bananas all the way down and consume them, for a reward of +10. Note that the other rewards are either effort-consuming (raising the hand or raising it and the bananas for rewards of -0.2 or -0.5) or somewhat easier, lowering the hand (for rewards 0f 0.1).



(a) (8 points) Create a table representing the function $T(s, a, s')$ for this MDP. Assume that there are three possible actions: U, D and T. U means try to go up, and D means try to go down. T means try to take the bananas (and go down from B to C). Action U is applicable only at states A, C, E, and F. Action D is applicable only at states A, B, C, and E. Action T is only applicable at state B. In this table you do not have to provide an explicit entry for any $T(s, a, s')$ whose value is 0. Hint: there should be 17 entries in your table.

s	**a**	**s'**	**prob**
A	U	B	0.8
A	U	F	0.2
A	D	F	0.8
A	D	B	0.2
B	D	A	0.8
B	D	C	0.2
B	T	C	0.8
B	T	A	0.2
C	U	B	0.8
C	U	E	0.2
C	D	E	0.8
C	D	B	0.2
E	U	C	0.8
E	U	F	0.2
E	D	F	0.8
E	D	C	0.2
F	U	A	1.0

(b) (5 points) Create a table representing the function $R(s, a, s')$. There should be one explicit entry here for each explicit entry you gave in part (a).

s	**a**	**s'**	**reward**
A	U	B	-0.2
A	U	F	0.1
A	D	F	0.1
A	D	B	-0.2
B	D	A	0.1
B	D	C	0.1
B	T	C	0.1
B	T	A	0.1
C	U	B	-0.5
C	U	E	0.1
C	D	E	0.1
C	D	B	-0.5
E	U	C	-0.5
E	U	F	10.0
E	D	F	10.0
E	D	C	-0.5
F	U	A	-0.2

(c) (8 points) Create a table representing the expected reward for taking action a in state s. Call this function $Q_1(s, a)$. Note: if action a is not allowed in state s then $S_1(s, a) = 0$. The expected reward is the *expectation* of the reward from state s given that the action is s. Because of the nondeterminism in the actions, this is normally a weighted sum of the possible rewards out of s, and the weight corresponding to action a is usually the largest of the weights. Hint: there should be 9 entries in your table.

s	**a**	**expected reward**
A	U	-0.2*0.8 + 0.1*0.2 = **-0.14**
A	D	0.1*0.8 - 0.2*0.2 = **0.04**
B	D	0.1*0.8 + 0.1*0.2 = **0.1**
B	T	0.1*0.8 + 0.1*0.2 = **0.1**
C	U	-0.5*0.8 + 0.1*0.2 = **-0.38**
C	D	0.1*0.8 - 0.5*0.2 = **-0.02**
E	U	-0.5*0.8 + 10.0*0.2 = **1.6**
E	D	10.0*0.8 - 0.5*0.2 = **7.9**
F	U	-0.2*1.0 = **-0.2**

(d) (4 points) What is the total reward corresponding to the following episode, where the monkey starts in state A and makes the transitions below?

```
A-D-F-U-A-U-B-D-A-U-B-T-C-D-B-T-C-D-E-D-C-D-E-D-F-U-A
```

(Keep in mind that the actions are generally nondeterministic.)

0.1 - 0.2 - 0.2 + 0.1 - 0.2 + 0.1 - 0.5 + 0.1 + 0.1 - 0.5 + 0.1 + 10 - 0.2 = 8.8

NOTE: These are sample responses. Free responses from students may differ and will need to be evaluated individually.