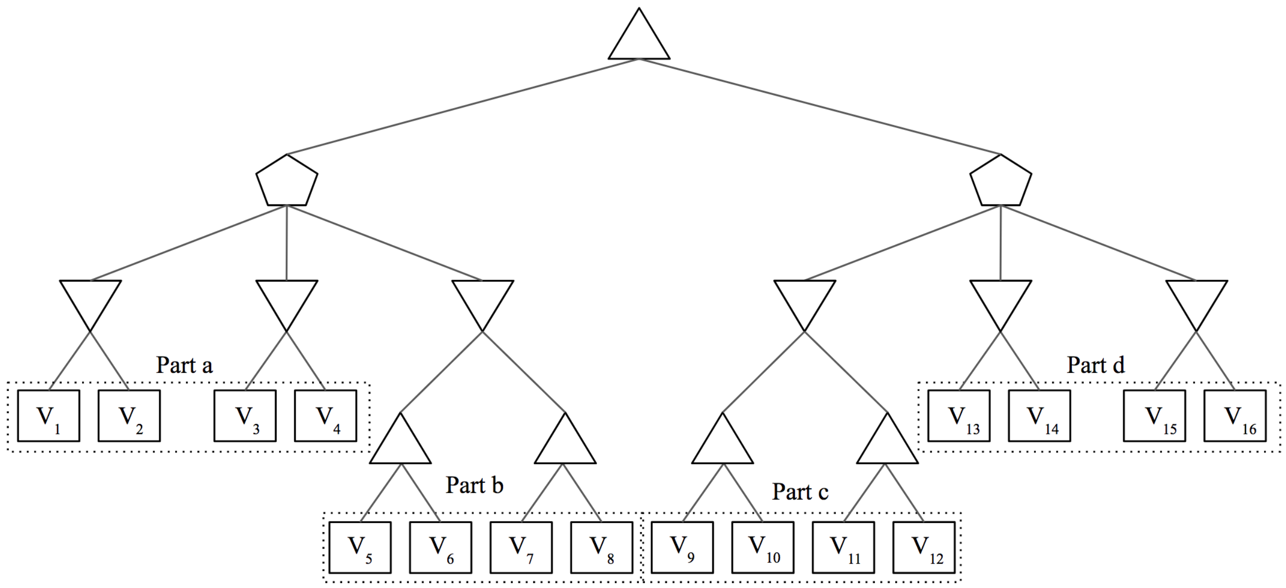
CS 188 Fall 2022 — Introduction to Artificial Intelligence

Exam Prep 4 Solutions

Q1. MedianMiniMax

You're living in utopia! Despite living in utopia, you still believe that you need to maximize your utility in life, other people want to minimize your utility, and the world is a 0 sum game. But because you live in utopia, a benevolent social planner occasionally steps in and chooses an option that is a compromise. Essentially, the social planner (represented as the pentagon) is a median node that chooses the successor with median utility. Your struggle with your fellow citizens can be modelled as follows:



There are some nodes that we are sometimes able to prune. In each part, mark all of the terminal nodes such that **there exists a possible situation** for which the node **can be pruned**. In other words, you must consider **all** possible pruning situations. Assume that evaluation order is **left to right** and all V_i's are **distinct.**

Note that as long as there exists ANY pruning situation (does not have to be the same situation for every node), you should mark the node as prunable. Also, alpha-beta pruning does not apply here, simply prune a sub-tree when you can reason that its value will not affect your final utility.

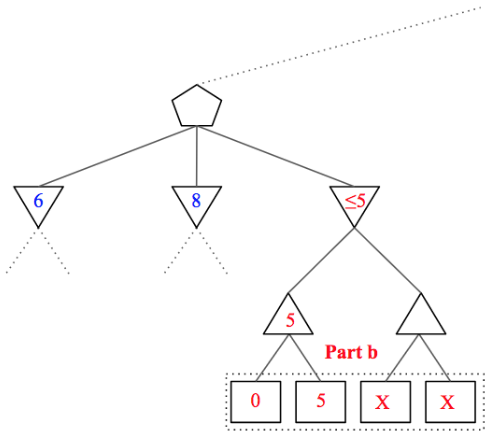
(a)
- ☐ V_1
- ☐ V_2
- ☐ V_3
- ☐ V_4
- [✓ None]

(b)
- ☐ V_5
- [✓ V_6]
- [✓ V_7]
- [✓ V_8]
- ☐ None

(c)
- ☐ V_9
- ☐ V_{10}
- [✓ V_{11}]
- [✓ V_{12}]
- ☐ None

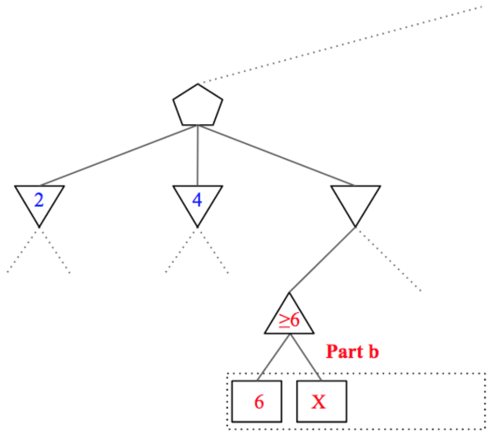
(d)
- ☐ V_{13}
- [✓ V_{14}]
- [✓ V_{15}]
- [✓ V_{16}]
- ☐ None

Part a:
For the left median node with three children, at least two of the childrens' values must be known since one of them will be guaranteed to be the value of the median node passed up to the final maximizer. For this reason, none of the nodes in part a can be pruned.



The value of this subtree will only get smaller.

The median node will **NOT** choose the value of this subtree. 6 is the median.



The value of this subtree will only get bigger.

*If the value of this subtree is chosen by the minimizer**, it will **NOT** be chosen by the median node.

**It is possible that the median is the value of the subtree to the right that we haven't looked at yet*

Part b (pruning V_7, V_8):
Let min_1, min_2, min_3 be the values of the three minimizer nodes in this subtree.

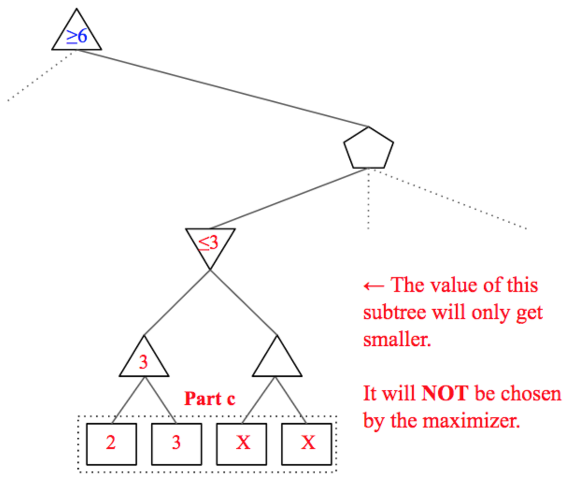
In this case, we may not need to know the final value min_3. The reason for this is that we may be able to put a bound on its value after exploring only partially, and determine the value of the median node as either min_1 or min_2 if $min_3 \leq \min(min_1, min_2)$ or $min_3 \geq \max(min_1, min_2)$.

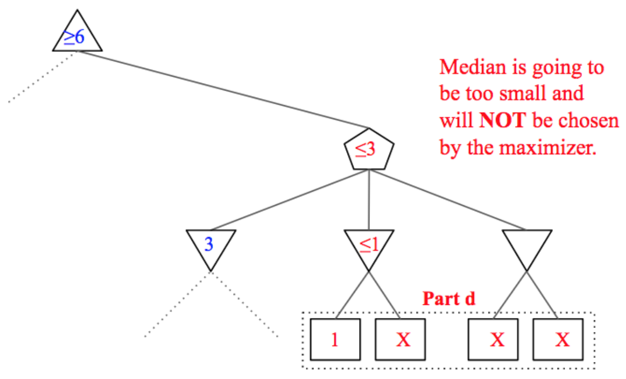
We can put an upper bound on min_3 by exploring the left subtree V_5, V_6 and if $\max(V_5, V_6)$ is lower than both min_1 and min_2, the median node's value is set as the smaller of min_1, min_2 and we don't have to explore V_7, V_8 in Figure 1.

Part b (pruning V_6):
It's possible for us to put a lower bound on min_3. If V_5 is larger than both min_1 and min_2, we do not need to explore V_6.

The reason for this is subtle, but if the minimizer chooses the left subtree, we know that $min_3 \geq V_5 \geq \max(min_1, min_2)$ and we don't need V_6 to get the correct value for the median node which will be the larger of min_1, min_2.

If the minimizer chooses the value of the right subtree, the value at V_6 is unnecessary again since the minimizer never chose its subtree.


≥6
≤3
3
Part c
2
3
X
X
← The value of this subtree will only get smaller.
It will NOT be chosen by the maximizer.


≥6
≤3
Median is going to be too small and will NOT be chosen by the maximizer.
3
≤1
Part d
1
X
X
X

Part c (pruning V_{11}, V_{12}):
Assume the highest maximizer node has a current value $max_1 \geq Z$ set by the left subtree and the three minimizers on this right subtree have value min_1, min_2, min_3.

In this part, if $min_1 \leq \max(V_9, V_{10}) \leq Z$, we do not have to explore V_{11}, V_{12}. Once again, the reasoning is subtle, but we can now realize if either $min_2 \leq Z$ or $min_3 \leq Z$ then the value of the right median node is for sure $\leq Z$ and is useless.

Only if both $min_2, min_3 \geq Z$ will the whole right subtree have an effect on the highest maximizer, but in this case the exact value of min_1 is not needed, just the information that it is $\leq Z$. Clearly in both cases, V_{11}, V_{12} are not needed since an exact value of min_1 is not needed.

We will also take the time to note that if $V_9 \geq Z$ we do have to continue the exploring as V_{10} could be even greater and the final value of the top maximizer, so V_{10} can't really be pruned.

Part d (pruning V_{14}, V_{15}, V_{16}):
Continuing from part c, if we find that $min_1 \leq Z$ and $min_2 \leq Z$ we can stop.

We can realize this as soon we explore V_{13}. Once we figure this out, we know that our median node's value must be one of these two values, and neither will replace Z so we can stop.

Q2. How do you Value It(eration)?

(a) Fill out the following True/False questions.

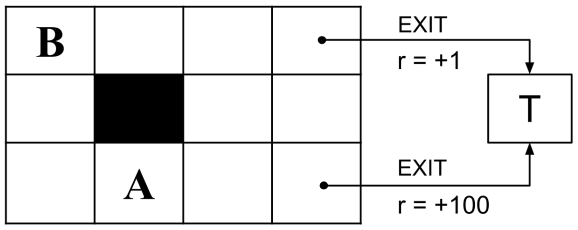
(i) [✓ True] ○ False: Let A be the set of all actions and S the set of states for some MDP. Assuming that $|A| \ll |S|$, one iteration of value iteration is generally faster than one iteration of policy iteration that solves a linear system during policy evaluation. One iteration of value iteration is $O(|S|^2|A|)$, whereas one iteration of policy iteration is $O(|S|^3)$, so value iteration is generally faster when $|A| \ll |S|$

(ii) ○ True [✓ False]: For any MDP, changing the discount factor does not affect the optimal policy for the MDP. Consider an infinite horizon setting where we have 2 states A, B, where we can alternate between A and B forever, gaining a reward of 1 each transition, or exit from B with a reward of 100. In the case that $\gamma = 1$, the optimal policy is to forever oscillate between A and B. If $\gamma = \frac{1}{2}$, then it is optimal to exit.

The following problem will take place in various instances of a grid world MDP. Shaded cells represent walls. In all states, the agent has available actions ↑, ↓, ←, →. Performing an action that would transition to an invalid state (outside the grid or into a wall) results in the agent remaining in its original state. In states with an arrow coming out, the agent has an *additional* action $EXIT$. In the event that the $EXIT$ action is taken, the agent receives the labeled reward and ends the game in the terminal state T. Unless otherwise stated, all other transitions receive no reward, and all transitions are deterministic.

For all parts of the problem, assume that value iteration begins with all states initialized to zero, i.e., $V_0(s) = 0 \; \forall s$. **Let the discount factor be $\gamma = \frac{1}{2}$ for all following parts.**

(b) Suppose that we are performing value iteration on the grid world MDP below.



(i) Fill in the optimal values for A and B in the given boxes.

$V^*(A)$: $\boxed{25}$ $V^*(B)$: $\boxed{\frac{25}{8}}$

(ii) After how many iterations k will we have $V_k(s) = V^*(s)$ for all states s? If it never occurs, write "never". Write your answer in the given box.

$\boxed{6}$

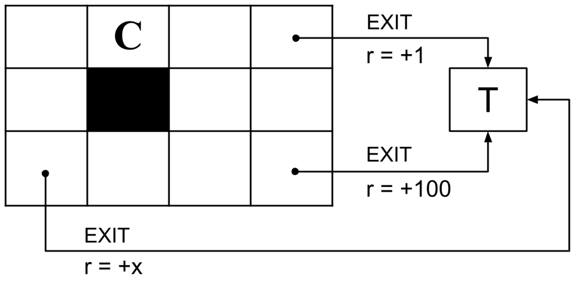
(iii) Suppose that we wanted to re-design the reward function. For which of the following new reward functions would the optimal policy **remain unchanged**? Let $R(s, a, s')$ be the original reward function.

- ■ $R_1(s, a, s') = 10R(s, a, s')$
- ■ $R_2(s, a, s') = 1 + R(s, a, s')$
- ■ $R_3(s, a, s') = R(s, a, s')^2$
- □ $R_4(s, a, s') = -1$
- □ None

R_1: Scaling the reward function does not affect the optimal policy, as it scales all Q-values by 10, which retains ordering
R_2: Since reward is discounted, the agent would get more reward exiting then infinitely cycling between states

R_3: The only positive reward remains to be from exiting state +100 and +1, so the optimal policy doesn't change
R_4: With negative reward at every step, the agent would want to exit as soon as possible, which means the agent would not always exit at the bottom-right square.

(c) For the following problem, we add a new state in which we can take the $EXIT$ action with a reward of $+x$.



(i) For what values of x is it *guaranteed* that our optimal policy π^* has $\pi^*(C) = \leftarrow$? Write ∞ and $-\infty$ if there is no upper or lower bound, respectively. Write the upper and lower bounds in each respective box.

$\boxed{50} < x < \boxed{\infty}$

We go left if $Q(C, \leftarrow) > Q(C, \rightarrow)$. $Q(C, \leftarrow) = \frac{1}{8}x$, and $Q(C, \rightarrow) = \frac{100}{16}$. Solving for x, we get $x > 50$.

(ii) For what values of x does value iteration take the **minimum** number of iterations k to converge to V^* for all states? Write ∞ and $-\infty$ if there is no upper or lower bound, respectively. Write the upper and lower bounds in each respective box.

$\boxed{50} \leq x \leq \boxed{200}$

The two states that will take the longest for value iteration to become non-zero from either $+x$ or $+100$, are states C, and D, where D is defined as the state to the right of C. C will become nonzero at iteration 4 from $+x$, and D will become nonzero at iteration 4 from $+100$. We must bound x so that the optimal policy at D does not choose to go to $+x$, or else value iteration will take 5 iterations. Similar reasoning for D and $+x$. Then our inequalities are $\frac{1}{8}x \geq \frac{100}{16}$ and $\frac{1}{16}x \leq \frac{100}{8}$. Simplifying, we get the following bound on x: $50 \leq x \leq 200$

(iii) Fill the box with value k, the **minimum** number of iterations until V_k has converged to V^* for all states.

$\boxed{4}$

See the explanation for the part above