Exam Review Questions (Search and CSP)

Search

Question 1: Search Problem

It is training day for Pacbabies, also known as Hungry Running Maze Games day. Each of k Pacbabies starts in its own assigned start location s_i in a large maze of size MxN and must return to its own Pacdad who is waiting patiently but proudly at g_i; along the way, the Pacbabies must, between them, eat all the dots in the maze.

At each step, all k Pacbabies move one unit to any open adjacent square. The only legal actions are Up, Down, Left, or Right. It is illegal for a Pacbaby to wait in a square, attempt to move into a wall, or attempt to occupy the same square as another Pacbaby. To set a record, the Pacbabies must find an optimal collective solution

a. Define a minimal state space representation for this problem

We need to keep track on k Pacbabies positions along with the food eaten. So, the state space is defined by the current location of k Pacbabies and for each of M*N position, Boolean variables indicating whether a food at location i,j is eaten or not

b. How large is the state space?

$MN^k\ 2^{MN}$

MN^k is for k Pacbabies locations

2^{MN} is for food positions

c. What is the maximum branching factor for this problem?

4^k. Each of the k Pacbabies can make one of the four actions

d. Let MH(p, q) be the Manhattan distance between positions p and q and F be the set of all positions of remaining food pellets and p_i be the current position of Pacbaby$_i$. Which of the following are admissible heuristics?

H1: $\frac{\sum_{i=1}^{k} MH(p_i,q_i)}{k}$

H2: $\max_{1 \le i \le k} MH(p_i, q_i)$

H3: $\max_{1 \le i \le k}(\max_{f\ in\ F} MH(p_i, f))$

H4: $\max_{1 \le i \le k} (\min_{f \ in \ F} MH(p_i, f))$

H5: $\min_{1 \le i \le k} (\max_{f \ in \ F} MH(p_i, f))$

H6: $\min_{1 \le i \le k} (\max_{f \ in \ F} MH(p_i, f))$

H1 is admissible because the total Pacbaby–Pacdad distance can be reduced by at most k at each time step

H2 is admissible because you need at least furthest Pacbaby to reach its Pacdad

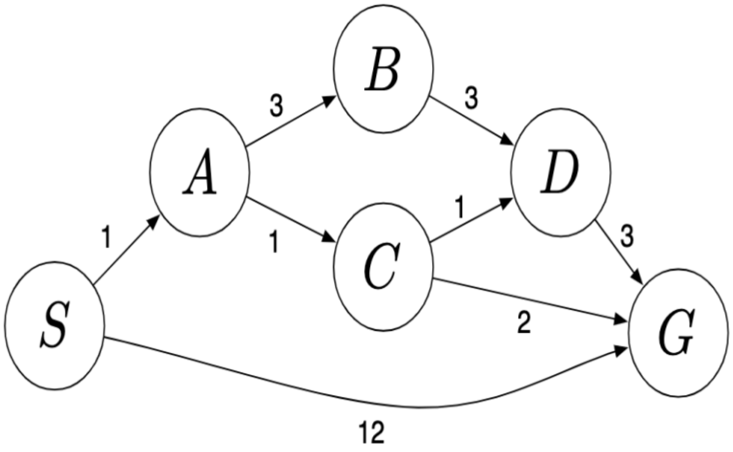
H3 is not admissible because it looks at the distance from each Pacbaby to its most distant food square; but of course, the optimal solution might another Pacbaby going to that square; same problem for H4

H5 is admissible because some Pacbaby will have to travel at least this far to eat one piece of food (but it's not very accurate)

H6 is not admissible because it connects each food square to the most distant Pacbaby, which may not be the one who eats it.

Question 2: Search Graph and Algorithms

In the figure below. Answer the following questions about the search problem shown above. Assume that ties are broken alphabetically. (For example, a partial plan S→X→A would be expanded before S→X→B; similarly, S→A→Z would be expanded before S→B→A.)



a. What path would breadth-first graph search return for this search problem?

S→ G

b. What path would uniform cost graph search return for this search problem?

S→A→C→G

c. What path would depth-first graph search return for this search problem?

S→A→B→D→G

d. What path would A* graph search, using a consistent heuristic, return for this search problem?

S→A→C→G

e. Consider the heuristics for this problem shown in the table below.

State	H1	H2
S	5	3
A	3	2
B	6	6
C	2	1
D	3	3
G	0	0

a. Is H1 admissible?

No. 5 (for S) is more than the actual cost (which is 4)

b. Is H1 consistent?

No. Not admissible won’t be consistent

c. Is H2 admissible?

Yes. Every path is less than the actual cost

d. Is H2 consistent?

No. Cost of S-A is 1 so H(S)-H(A)=3-1=2 so H(S)-H(A)> Cost (S→A). Inconsistent!

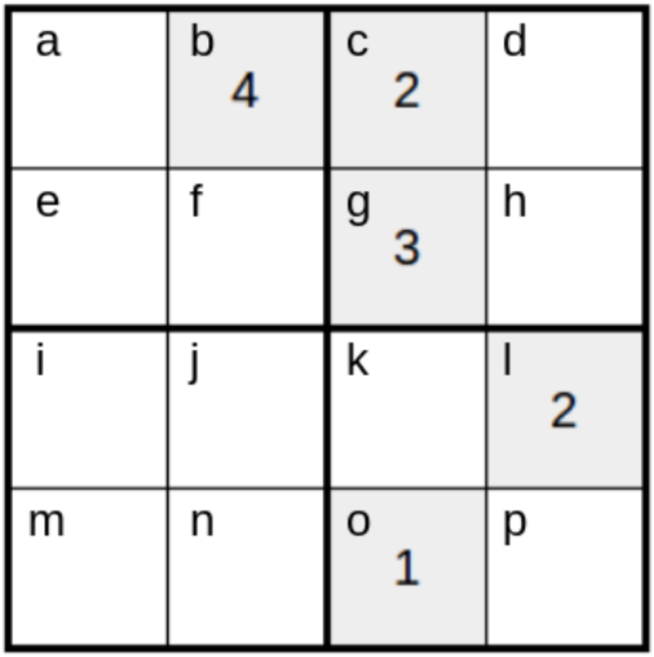
e. Propose a heuristic H3 that is consistent

State	H3
S	2
A	2
B	6
C	1
D	3
G	0

CSP

Question 3 CSP

Pacman is playing a simplified version of a Sudoku puzzle. The board is a 4-by-4 square, and each box can have a number from 1 through 4. In each row and column, a number can only appear once. Furthermore, in each group of 2-by-2 boxes outlined with a solid border, each of the 4 numbers may only appear once as well. For example, in the boxes a, b, e, and f, each of the numbers 1 through 4 may only appear once. Note that the diagonals do not necessarily need to have each of the numbers 1 through 4. In front of Pacman, he sees the board below. Notice that the board already has some boxes filled out! Box b= 4, c= 2, g= 3, l= 2, and o= 1.



a	b 4	c 2	d
e	f	g 3	h
i	j	k	l 2
m	n	o 1	p

Explicitly, we represent this simple Sudoku puzzle as a CSP which has the constraints

1. Each box can only take on values 1, 2, 3, or 4.
2. 1, 2, 3, and 4 may only appear once in each row.
3. 1, 2, 3, and 4 may only appear once in each column.
4. 1, 2, 3, and 4 may only appear once in each set of 2-by-2 boxes with solid borders.
5. b= 4, c= 2, g= 3, l= 2, and o= 1.

A. Pacman is very excited and decides to naively do backtracking using only forward checking. Assume that he solves the board from left to right, top to bottom (so he starts with box a and proceeds to box b, then c, etc.), and assume that he has already enforced all unary constraints. Pacman assigns 3 to box a. If he runs forward checking, which boxes' domains should he attempt to prune?

In forward checking, we only prune the domains of variables that the assignment directly affects. Looking at the constraints, these are the ones that are in the same row/column, and the ones that are in its 2-by-2 grid. So, d, e, f, i, and m.

B. Pacman decides to start over and play a bit smarter. He now wishes to use arc-consistency to solve this Sudoku problem. Assume for all parts of the problem Pacman has already enforced all unary constraints, and Pacman has erased what was previously on the board.

a. How many arcs are there in the queue prior to assigning any variables or enforcing arc consistency?

(4+4)*4*3 + (4*4) = 112. There are 4 rows, 4 columns each with 4 permute 2 arcs. Additionally, for each of the 4 2-by-2 boxes, there are the 4 diagonals. Another way to think about it is that each variable will have 7 arcs (horizontal and vertical only and there are 16 variables.

b. Enforce the arc d→c, from box d to box c. What are values remaining in the domain of box d?

1,3,4

By the constraint that no two boxes in the same row can have the same number, after enforcing this arc we know that d cannot be 2. We cannot say anything else just from enforcing this single arc

c. After enforcing arc consistency, what is the domain of each box in the first row?

a 3

b 4

c 2

d 1

Because of unary constraints, b=4 and c=2 (domains: b={4}, c={2}. By enforcing arcs (specifically d→c, d→g, and d→l), we know that d has to 1. Similarly, by constraining arcs a→b, a→c, and a→d, we get that the domain of a can only be 3.

Question 4 CSP-Scheduling (Time Management)

The TA and the instructor are making their schedules for a busy morning. There are five tasks to be carried out:

(F) Pick up food for the group's research seminar, which, sadly, takes one precious hour.

(H) Prepare homework questions, which takes 2 consecutive hours.

(P) Prepare some research slides for a group of preschoolers' visit, which takes one hour.

(S) Lead the research seminar, which takes one hour.

(T) Teach the preschoolers about the research, which takes 2 consecutive hours.

The schedule consists of one-hour slots:

8am-9am, 9am-10am, 10am-11am, 11am-12pm.

The requirements for the schedule are as follows:

1. In any given time slot, each one can do at most one task (F, H, P, S, T).
2. The research preparation (P) should happen before teaching the preschoolers (T).
3. The food should be picked up (F) before the seminar (S).
4. The seminar (S) should be finished by 10am.
5. The instructor is going to deal with food pick up (F) since he has a car.
6. The one not leading the seminar (S) should still attend, and hence cannot perform another task (F, T, P,H) during the seminar.
7. The seminar (S) leader does not teach the preschoolers (T).
8. The one who teaches the preschoolers (T) must also prepare the presentation (P).
9. Preparing homework questions (H) takes 2 consecutive hours, and hence should start at or before 10am.
10. Teaching the preschoolers (T) takes 2 consecutive hours, and hence should start at or before 10am.

To formalize this problem as a CSP, use the variables F, H, P, S and T. The values they take on are: TA or instructor, indicating the person responsible for it, and the starting time slot during which the task is carried out (for a task that spans 2 hours, the variable represents the starting time, but keep in mind that the person responsible will be occupied for the next hour also - make sure you enforce constraint). Hence there are eight possible values for each variable, which we will denote by S8, S9, S10, S11, C8, C9, C10, C11, where the letter corresponds to the person (S for instructor, C is for TA) and the number corresponds to the time slot. For example, assigning the value of A8 to a variables means that this task is carried about by the instructor from 8am to 9am.

i. What is the size of the state space for this CSP?

8^5. 8 values (S8, S9, S10, S11, C8, C9, C10, C11) and 5 variables (F, H, P, S and T). In other words, d^n where d is the length of the domain (i.e., 8) and n is the number of variables (i.e., 5)

ii. Which of the statements above include unary constraints?

4, 5,9, 10. Reason: unary constrains are those that involve one variable.

iii. In the table below, enforce all unary constraints by crossing out values in the table below.

F	S8	S9	S10	S11	C8	C9	C10	C11
H	S8	S9	S10	S11	C8	C9	C10	C11
P	S8	S9	S10	S11	C8	C9	C10	C11
S	S8	S9	S10	S11	C8	C9	C10	C11
T	S8	S9	S10	S11	C8	C9	C10	C11

Solution

F	S8	S9	S10	S11	C8	C9	C10	C11
H	S8	S9	S10	S11	C8	C9	C10	C11
P	S8	S9	S10	S11	C8	C9	C10	C11
S	S8	S9	S10	S11	C8	C9	C10	C11
T	S8	S9	S10	S11	C8	C9	C10	C11

Enforcing 4, 5,9, 10 directly.

Notations: Blue is left domain, red is crossed from the domain, green is assigned

You will have such questions in the exam. Make sure that you try to solve them as correct as possible. It is really hard to give partial credits for these. Also, other parts will depend on these and it will be wrong if these are wrong.

iv. Start from the table above, select the variable S and assign the value S9 to it. Perform forward checking by crossing out values in the table

Solution:

F	S8	S9	S10	S11	C8	C9	C10	C11
H	S8	S9	S10	S11	C8	C9	C10	C11

P	S8	S9	S10	S11	C8	C9	C10	C11
S	S8	S9	S10	S11	C8	C9	C10	C11
T	S8	S9	S10	S11	C8	C9	C10	C11

Enforcing direct constraint 1,3,4,6, and 7 is easy (direct). Tricky part with H and T. They should be hosted for 2 hours, and now we know that it cann't be at 9 because both persons will be busy with the seminar. So, 8 should be crossed for both S and C (i.e., S8, C8 should be crossed from H and T)

v. Based on the result of (d), what variable will we choose to assign next based on the MRV heuristic (breaking ties alphabetically)? Assign the first possible value to this variable, and perform forward checking by crossing out values in the table below.

Variables selected is ------ and gets assigned value ---- .

Have we arrived at a dead end (i.e., has any of the domains become empty)?

Solution:

F	S8	S9	S10	S11	C8	C9	C10	C11
H	S8	S9	S10	S11	C8	C9	C10	C11
P	S8	S9	S10	S11	C8	C9	C10	C11
S	S8	S9	S10	S11	C8	C9	C10	C11
T	S8	S9	S10	S11	C8	C9	C10	C11

Variables selected is ---F--- and gets assigned value ---S8- .

Have we arrived at a dead end (i.e., has any of the domains become empty)? No.

vi. We return to the result from enforcing just the unary constraints, which we did in (c). Select the variable S and assign the value A9. Enforce arc consistency by crossing out values in the table below.

Solution:

F	S8	S9	S10	S11	C8	C9	C10	C11
H	S8	S9	S10	S11	C8	C9	C10	C11
P	S8	S9	S10	S11	C8	C9	C10	C11
S	S8	S9	S10	S11	C8	C9	C10	C11
T	S8	S9	S10	S11	C8	C9	C10	C11

Enforce arc consistency will lead to these domains.

vii. Compare your answers to (d) and to (f). Does arc consistency remove more values or less values than forward checking does? Explain why

Arc consistency removes more values. It’s because AC checks consistency between any pair of variables, while FC only checks the relationship between pairs of assigned and unassigned variables

viii. Check your answer to (f). Without backtracking, does any solution exist along this path? Provide the solution(s) or state that there is none.

AC along this path gives 1 solution: F=A8, H=A10, P=D8, S=A9, T=D10

Games

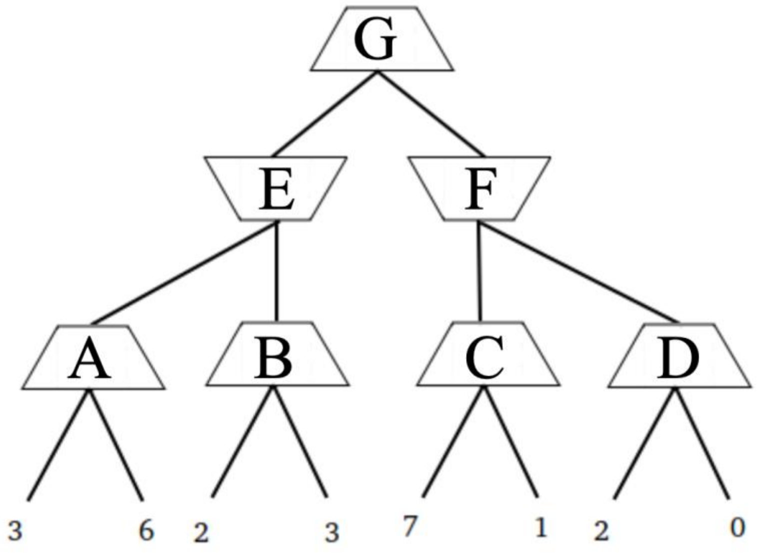
Question 5: Games with Magic

a. Standard Minimax

i. Fill in the values of each of the nodes in the following Minimax tree. The upward pointing trapezoids correspond to maximizer nodes (layer 1 and 3), and the downward pointing trapezoids correspond to minimizer nodes (layer 2). Each node has two actions available, Left and Right.

A: B: C: D: E: F: G:

ii. State the sequence of actions that correspond to Minimax play (for example, G→F→D→0



Solution:

i. A: 6 B:3 C:7 D:2 E:3 F:2 G:3

ii. G→E→B→3

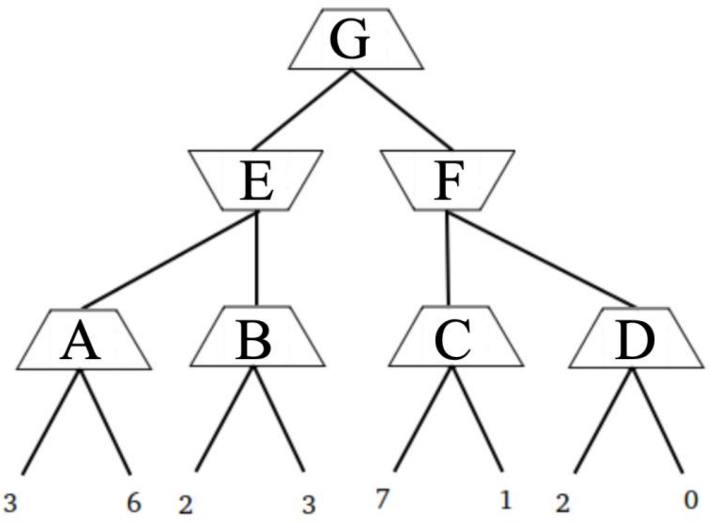
b. Dark Magic

Pacman (= maximizer) has mastered some dark magic. With his dark magic skills Pacman can take control over his opponent’s muscles while they execute their move — and in doing so be fully in charge of the opponent’s move. But the magic comes at a price: every time Pacman uses his magic, he pays a price of c—which is measured in the same units as the values at the bottom of the tree.

Note: For each of his opponent's actions, Pacman has the choice to either let his opponent act (optimally according to minimax), or to take control over his opponent's move at a cost of c.

i. Dark Magic at Cost c= 2

Consider the same game as before but now Pacman has access to his magic at cost c= 2. Fill in the tree below and choose the optimal path for Pacman. Is it optimal for Pacman to use his dark magic? Either way, state what the outcome of the game and the sequence of actions that lead to that outcome.



A: B: C: D: E: F: G:

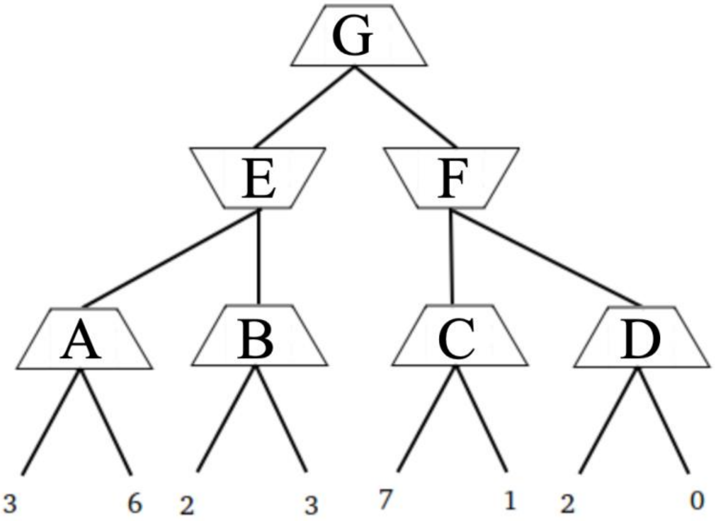
Solution:

A:6 B:3 C:7 D:2 E:4 F:5 G:5

Outcome is 5. Sequence is G→F→C→7

ii. Dark Magic at Cost c= 2

Consider the same game as before but now Pacman has access to his magic at cost c= 5. Fill in the tree below and choose the optimal path for Pacman. Is it optimal for Pacman to use his dark magic? Either way, state what the outcome of the game and the sequence of actions that lead to that outcome.



A: B: C: D: E: F: G

Solution:

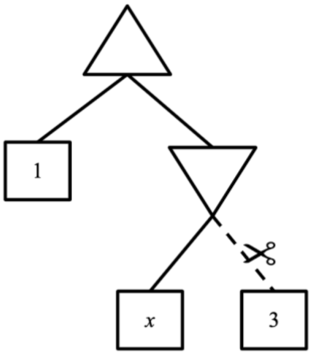
A:6 B:3 C:7 D:2 E:3 F:2 G:3

Outcome is 3. Sequence is G→E→B→3. Pacman doesn't use dark magic

Question 6: Games: Alpha-Beta Pruning

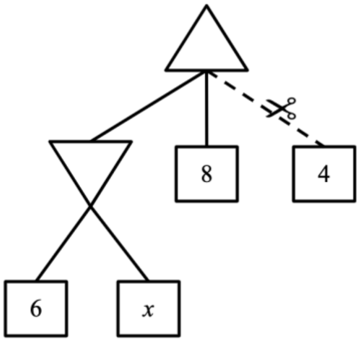
For each of the game-trees shown below, state for which values of x the dashed branch with the scissors will be pruned. If the pruning will not happen for any value of x write "none". If pruning will happen for all values of x write "all".

a. x?



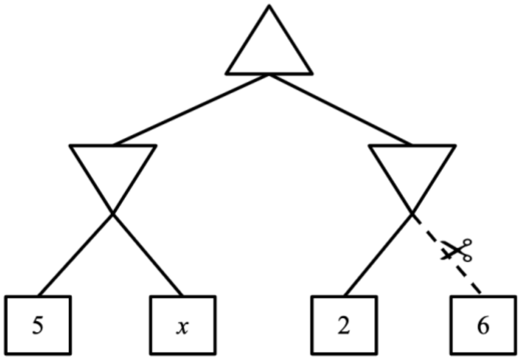
Answer: x<=1

b. x?



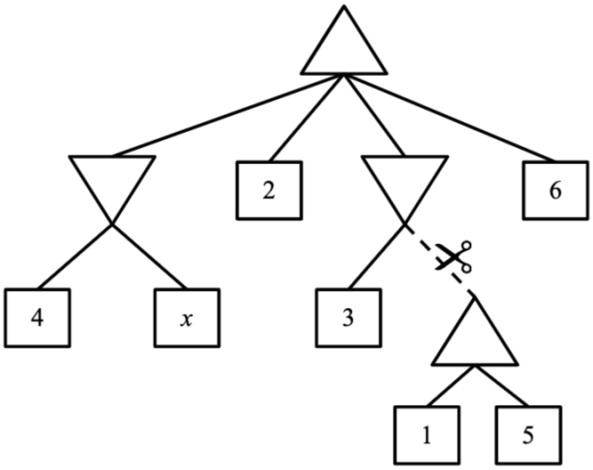
Answer: None

c. x?



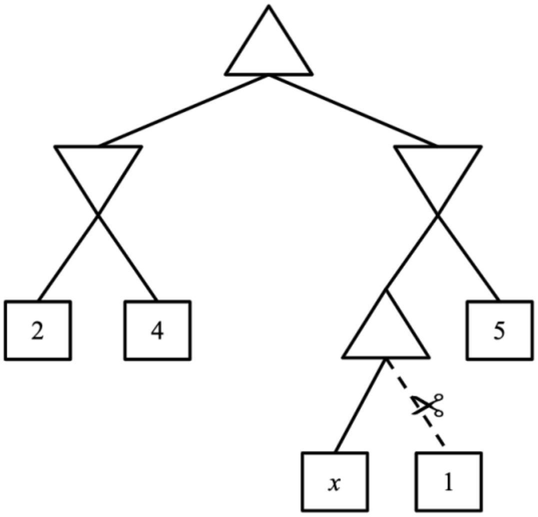
Answer: x>=2

d. x?



Answer: x>=3

e. x?

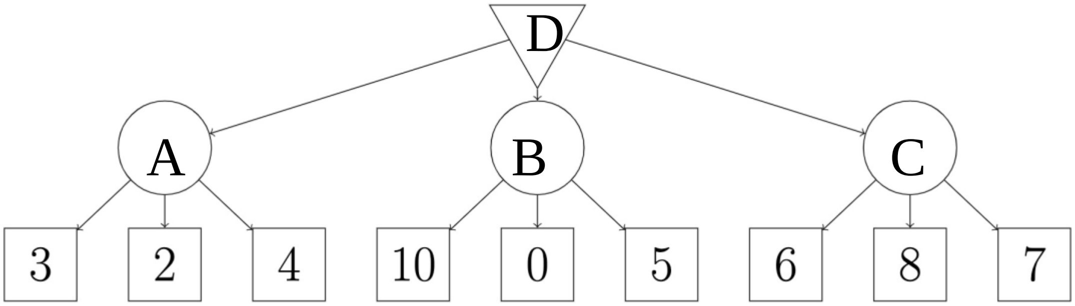


Answer: none

Question 7: Expectimin

In this problem we model a game with a minimizing player and a random player. We call this combination “expectimin” to contrast it with expectimax with a maximizing and random player. Assume all children of expectation nodes have equal probability and sibling nodes are visited left to right for all parts of this question

a. Fill out the “expectimin” tree below



A: B: C: D:

Solution:

A:3 B:5 C:7 D:3

b. Suppose that before solving the game we are given additional information that all values are non-negative and all nodes have exactly 3 children. Which leaf nodes in the tree above can be pruned with this information

3 2 4 10 0 5 6 8 7

Solution:

3 2 4 10 **0** **5** 6 8 **7**

The branch from A is 3, i.e., A<=3. Then, we see 10 from the second branch. So, even if the other two branches are 0, the average is 3.3→ more than 3 and won't be chosen. The third branch, first we see 6. If the other two branches are 0's, the average is less than 3. So, only after seeing 8 we can prune.

c. In which of the following other games can we also use some form of pruning?

i. Expectimax

ii. Expectimin

iii. Expectimax with all non-negative values and known number of children

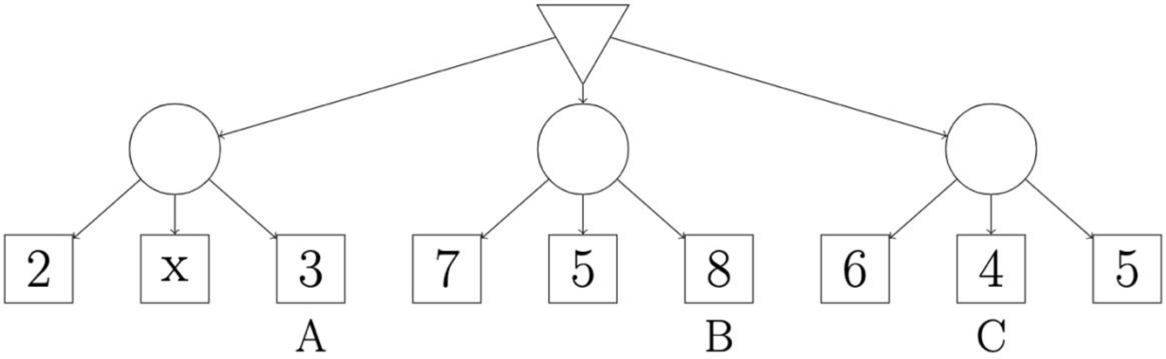
iv. Expectimax with all non-positive values and known number of children

v. Expectimin with all non-positive values and known number of children

Solution:

Choice is iv. Expectimax and expectimin can't be pruned in general since any single child of an expectation node can arbitrarily change its value. Expectimax has maximizer nodes that will accumulate lower bounds, so the value ranges must help us give upper bounds on the expectations, which means the values must be bounded from above. Expectimin with non-positive values does not allow pruning since both the minimizer and expectation nodes will accumulate upper bounds

d. For each of the leaves labeled A, B, and C in the tree below, determine which values of x will cause the leaf to be pruned, given the information that all values are non-negative and all nodes have 3 children. Assume we do not prune on equality



i. A: X< X> none any

Solution:

X< X> none any

No solution will ever prune from the first branch. It should always be completed.

ii. B: X< X> none any

Solution:

$X<7$ X> none any

To prune B, the value of the first expectation node must be less than the value of the second even if B were 0.

So $(2+X+3)/3 <(7+5)/3$ → $(5+X)/3 <4$ → $X<7$

iii. C: X< X> none any

Solution:

$X<1$ X> none any

To prune B, the value of the first expectation node must be less than the value of the second even if B were 0.

So $(2+X+3)/3 <6/3$ → $(5+X)/3 <2$ → $X<1$

MDP

Question 8: Micro-Blackjack

In micro-blackjack, you repeatedly draw a card (with replacement) that is equally likely to be a 2, 3, or 4. You can either Draw or Stop if the total score of the cards you have drawn is less than 6. If your total score is 6 or higher, the game ends, and you receive a utility of 0. When you Stop, your utility is equal to your total score (up to 5), and the game ends. When you Draw, you receive no utility. There is no discount (γ= 1). Let's formulate this problem as an MDP with the following states: 0,2,3,4,5 and a Done state, for when the game ends

1. What is the transition function and the reward function for this MDP?

Solution:

The transition function is

T (s, Stop, Done) = 1

T (0, Draw, s′) = 1/3 for s′∈ {2, 3, 4}

T (2, Draw, s′) = 1/3 for s′∈ {4, 5, Done}

T (3, Draw, s′) =1/3 if s′= 5 or 2/3 if s′=Done

T (4, Draw, Done) = 1

T (5, Draw, Done) = 1

T (s, a, s′) = 0 otherwise

The reward function is

R(s, Stop, Done) =s, s≤5

R(s, a, s′) = 0 otherwise

2. Fill in the following table of value iteration values for the first 4 iterations.

State	0	2	3	4	5
V0					
V1					
V2					
V3					
V4					

Solution:

State	0	2	3	4	5

V0	0	0	0	0	0
V1	0	2	3	4	5
V2	3	3	3	4	5
V3	10/3	3	3	4	5
V4	10/3	3	3	4	5

3. You should have noticed that value iteration converged above. What is the optimal policy for the MDP?

State	0	2	3	4	5
Optimal policy π*					

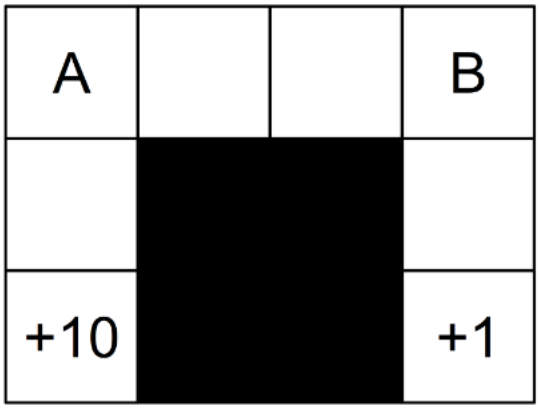
Solution:

State	0	2	3	4	5
Optimal policy π*	Draw	Draw	Stop	Stop	Stop

Question 9. Value Iteration

An agent lives in gridworld G consisting of grid cells s∈S, and is not allowed to move into the cells colored black. In this gridworld, the agent can take actions to move to neighboring squares, when it is not on a numbered square. When the agent is on a numbered square, it is forced to exit to a terminal state (where it remains), collecting a reward equal to the number written on the square in the process.

Gridworld *G*



You decide to run value iteration for gridworld G. The value function at iteration k is $V_k(s)$. The initial value for all grid cellsis 0 (that is,$V_0(s) = 0$ for all $s \in S$). When answering questions about iteration k for $V_k(s)$, either answer with a finite integer or ∞. For all questions, the discount factor is $\gamma = 1$.

1. Consider running value iteration in gridworld G. Assume all legal movement actions will always succeed (and so the state transition function is deterministic)

   a. What is the smallest iteration k for which $V_k(A)>0$? For this smallest iteration k, what is the value $V_k(A)$?

   k= $V_k(A)=$

Solution

k=3 $V_k(A)=10$

The nearest reward is10, which is 3 steps away. Because $\gamma = 1$, there is no decay in the reward, so the value propagated is 10

   b. What is the smallest iteration k for which $V_k(B)>0$? For this smallest iteration k, what is the value $V_k(B)$?

   k= $V_k(B)=$

Solution

k=3 $V_k(B)=1$

The nearest reward is1, which is3steps away. Because $\gamma = 1$, there is no decay in the reward, so the value propagated is 10

   c. What is the smallest iteration k for which $V_k(A)= V_k^*(A)$? What is the value of $V_k^*(A)$?

   k= $V_k^*(A)=$

Solution

k=3 $V_k^*(A)=1$

Because $\gamma = 1$, the problem reduces to finding the distance to the highest reward (because there is no living reward). The highest reward is10, which is 3 steps away.

   d. What is the smallest iteration k for which $V_k(B)= V_k^*(B)$? What is the value of $V_k^*(B)$?

   k= Vk*(B)=

Solution

k=6 Vk*(B)=10

Because γ= 1, the problem reduces to finding the distance to the highest reward (because there is no living reward). The highest reward is10, which is 6 steps away.

e. Now assume all legal movement actions succeed with probability 0.8; with probability 0.2, the action fails and the agent remains in the same state. Consider running value iteration in gridworld G. What is the smallest iteration k for which $V_k(A) = V_k^*(A)$? What is the value of $V_k^*(A)$?

   k=

   $V_k^*(A)$=

Solution

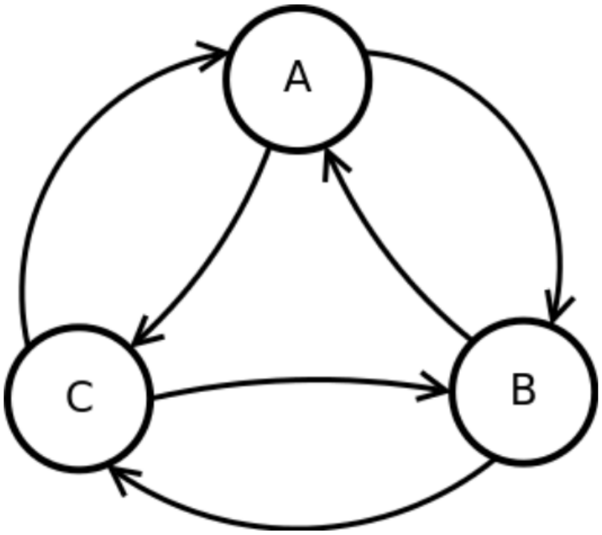
k= inf

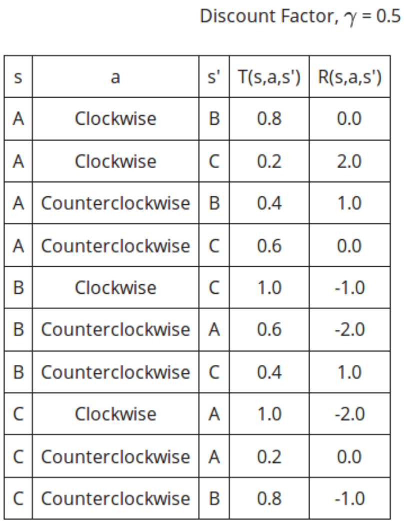
$V_k^*(A)$= 10

Because γ= 1 and the only rewards are in the exit states, the optimal policy will move to the exit state with highest reward. This is guaranteed to ultimately succeed, so the optimal value of state A is 10. However, because the transition is non-deterministic, it's not guaranteed this reward can be collected in 3 steps. It could any number of steps from 3 through infinity, and the values will only have converged after infinitely many iterations.

Question 10: Policy Iteration: Cycle

Consider the following transition diagram, transition function and reward function for an MDP.

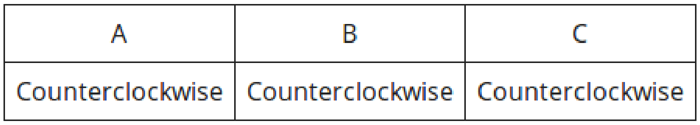




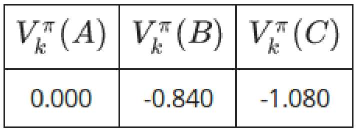
Discount Factor, $\gamma = 0.5$

s	a	s'	T(s,a,s')	R(s,a,s')
A	Clockwise	B	0.8	0.0
A	Clockwise	C	0.2	2.0
A	Counterclockwise	B	0.4	1.0
A	Counterclockwise	C	0.6	0.0
B	Clockwise	C	1.0	-1.0
B	Counterclockwise	A	0.6	-2.0
B	Counterclockwise	C	0.4	1.0
C	Clockwise	A	1.0	-2.0
C	Counterclockwise	A	0.2	0.0
C	Counterclockwise	B	0.8	-1.0

1. Suppose we are doing policy evaluation, by following the policy given by the left-hand side table below. Our current estimates (at the end of some iteration of policy evaluation) of the value of states when following the current policy is given in the right-hand side table.



A	B	C
Counterclockwise	Counterclockwise	Counterclockwise



$V_k^\pi(A)$	$V_k^\pi(B)$	$V_k^\pi(C)$
0.000	-0.840	-1.080

What is $V_{k+1}{}^{\pi}(A)$?

$V_{k+1}{}^{\pi}(A)=$

Solution

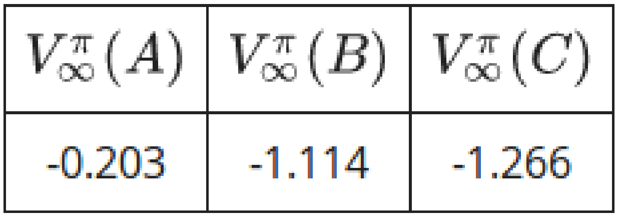
$V_{k+1}{}^{\pi}(A)= -.092$

Why?

$V_{k+1}{}^{\pi}(A)=T(A,\text{counterclockwise},B)[R(A,\text{counterclockwise},B)+ \gamma\ V_k{}^{\pi}(B)]+ T(A,\text{counterclockwise},C)[R(A,\text{counterclockwise},C)+\gamma\ V_k{}^{\pi}(C)]=-.092$

We only take into account the counterclockwise action from A, because that is the action according to our policy

2. Suppose that policy evaluation converges to the following value function, $V_\infty{}^\pi$



$V_\infty^\pi(A)$	$V_\infty^\pi(B)$	$V_\infty^\pi(C)$
-0.203	-1.114	-1.266

Now let's execute policy improvement.

a. What is Q_∞^π(A, clockwise)?

Q_∞^π(A, clockwise)=

Solution

Q_∞^π(A, clockwise) =-.1722

Why? Calculate Q_∞^π(A, clockwise)

Q_∞^π(A, clockwise)= T(A,clockwise,B)[R(A,clockwise,B)+γV_∞^π(B)]+ T(A,clockwise,C)[R(A,clockwise,C)+γV_∞^π(C)]]=−.1722

b. What is Q_∞^πA, counterclockwise)?

Q_∞^π(A, counterclockwise)=

Solution

Q_∞^π(A, counterclockwise)= -.2026

Same as before. Calculate Q_∞^π(A, counterclockwise)

c. What is the updated action for state A?

Solution

Clockwise. The updated action for state A will be the action that results in the higher Q