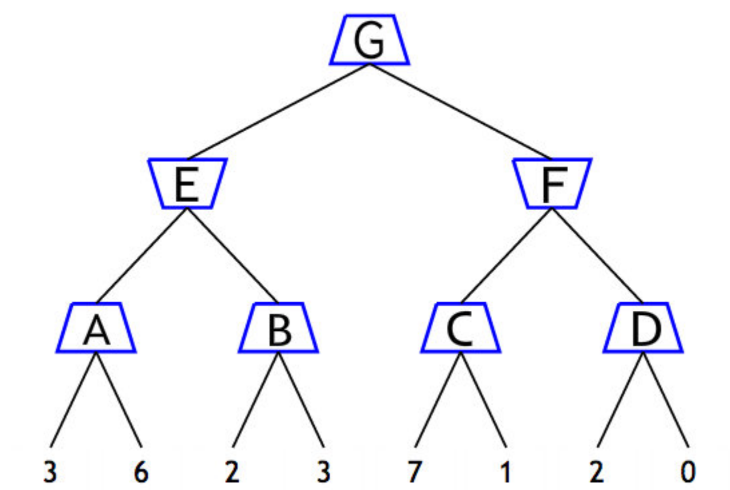
Q4: Games With Magic

Problem 4: Games with Magic

Part 1

3/3 points (ungraded)

Find the values of each of the letter nodes in the following Minimax tree, . The upward pointing trapezoids correspond to the maximizer nodes (layers 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
6	3	7	2	3	2
Correct!	Correct!	Correct!	Correct!	Correct!	Correct!

Submit

Correct (3/3 points)

Part 2

3/3 points (ungraded)
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.

Consider the same game as before but now Pacman has access to his magic at cost $c = 2$. What is the value of the game to Pacman?

5 ✔

What sequence of actions will lead to that value?

- ○ Left → Left → Right
- ○ Left → Right → Right
- ○ Left → Left → Left
- ◉ Right → Left → Left ✔
- ○ Right → Right → Left

Submit

✔ Correct (3/3 points)

Part 3

3/3 points (ungraded)
Consider the same game as before but now Pacman has access to his magic at cost $c = 5$. What is the value of the game to Pacman?

3 ✔

What sequence of actions will lead to that value?

- ○ Left → Left → Right
- ◉ Left → Right → Right ✔
- ○ Left → Left → Left
- ○ Right → Left → Left
- ○ Right → Right → Left

Submit

✔ Correct (3/3 points)

Part 4

7/7 points (ungraded)

Now let's study a general case. **Assume that the minimizer player has no idea that Pacman has the ability to use dark magic at a cost of c. I.e., the minimizer chooses their actions according to standard minimax.** We've given you partially complete pseudo-code used to compute Pacman's optimal value. Select the option below that completes the pseudo-code and correctly computes the optimal value.

v_{min} = the value of a node according to the minimizer's way of evaluating the game tree (unaware of the maximizer's ability to use magic (at a price))
v_{max} = the value of a node according to the maximizer's way of evaluating the game tree --- the maximizer does account for the ability to use magic, and does account for how the minimizer does their evaluations (and hence how the minimizer would choose their actions)

function $\textsc{Max-Value}(\text{state})$
  if *state* is leaf **then**
    return $(\textsc{Utility}(\text{state}),\ \textsc{Utility}(\text{state}))$
  end if
  $v_{min} \leftarrow -\infty$
  $v_{max} \leftarrow -\infty$
  for *successor* in $\textsc{Successors}(\text{state})$ **do**
    $vNext_{min}, vNext_{max} \leftarrow \textsc{Min-Value}(successor)$
    $v_{min} \leftarrow \max(v_{min}, vNext_{min})$
    $v_{max} \leftarrow \max(v_{max}, vNext_{max})$
  end for
  return $(v_{min},\ v_{max})$
end function

function $\textsc{Min-Value}(\text{state})$
  if *state* is leaf **then**
    return $(\textsc{Utility}(\text{state}),\ \textsc{Utility}(\text{state}))$
  end if
  $v_{min} \leftarrow \infty$
  $min\_move\_v_{max} \leftarrow -\infty$
  $v_{magic\_max} \leftarrow -\infty$
  for *successor* in $\textsc{Successors}(\text{state})$ **do**
    $vNext_{min}, vNext_{max} \leftarrow \textsc{Max-Value}(successor)$
    $\boxed{\qquad\qquad}$ (1)
    $v_{magic\_max} \leftarrow \max(vNext_{max}, v_{magic\_max})$
  end for
  $\boxed{\qquad\qquad}$ (2)
  return $(v_{min},\ v_{max})$
end function

○

(1)	(2)

`if` $v_{min} > vNext_{min}$ `then`
  $v_{min} \leftarrow vNext_{min}$
  $min\_move\_v_{max} \leftarrow vNext_{max} - c$
`end if`
	$v_{max} \leftarrow \max(min\_move\_v_{max},\ v_{magic\_max})$

- (selected)

if $v_{min} > vNext_{min}$ **then**
$v_{min} \leftarrow vNext_{min}$
$min\_move\_v_{max} \leftarrow vNext_{max}$
end if
	$v_{max} \leftarrow$max($min\_move\_v_{max}$, $v_{magic\_max} - c$)

(1)	(2)

✔

-

if $v_{magic\_max} < vNext_{max}$ **then**
$v_{min} \leftarrow vNext_{min}$
$min\_move\_v_{max} \leftarrow vNext_{max}$
end if
	$v_{max} \leftarrow$max($min\_move\_v_{max}$, $v_{magic\_max} - c$)

(1)	(2)

-

if $v_{min} > vNext_{min}$ **then**
$v_{min} \leftarrow vNext_{min}$
end if
	$v_{max} \leftarrow$max(v_{min}, $v_{magic\_max} - c$)

(1)	(2)

Submit

✔ Correct (7/7 points)

Part 5

7/7 points (ungraded)

The minimizer has come to the realization that Pacman has the ability to apply dark magic at cost c. Hence the minimizer now doesn't play according to the regular minimax strategy anymore, but accounts for Pacman's magic capabilities when making decisions. Pacman in turn, is also aware of the minimizer's new way of making decisions.

When Pacman uses dark magic at price c, not only is Pacman's utility decreased by c, the minimizer's utility (which is the negative of the leaf values of the game tree) is increased by c.

This time we've given you pseudo-code that computes the standard minimax value. Select the option to replace what's in the boxes and correctly compute Pacman's optimal value given the new situation.

```
function MAX-VALUE(state)
    if state is leaf then
        return UTILITY(state)
    end if
    v ← −∞
    for successor in SUCCESSORS(state) do
        v ← max(v, MIN-VALUE(successor))
    end for
    return v
end function
```

(1)

```
function MIN-VALUE(state)
    if state is leaf then
        return UTILITY(state)
    end if
    v ← ∞
    for successor in SUCCESSORS(state) do
        v ← min(v, MAX-VALUE(successor))
    end for
    return v
end function
```

(2)

---

(1)

```
for succ in SUCCESSORS (state) do
   v ←max(v, MIN-VALUE (succ))
end for
return v
```

(2)

```
v_m ← − inf
for succ in SUCCESSORS(state) do
   temp ← MAX-VALUE (succ)
   v ← min(v, temp)
   v_m ←max(v_m, temp)
end for
return max(v, v_m − c)
```

✔

---

(1)

```
v_m ← − inf
for succ in SUCCESSORS (state) do
   v ←max(v, MIN-VALUE (succ))
   v_m ←max(v_m, MAX-VALUE (succ))
end for
return max(v, v_m − c)
```

(2)

```
v_m ← − inf
for succ in SUCCESSORS(state) do
   temp ← MAX-VALUE (succ)
   v ← min(v, temp)
   v_m ←max(v_m, temp)
end for
return max(v, v_m − c)
```

$v_m \leftarrow -\inf$
for $succ$ in SUCCESSORS $(state)$ do
  $v \leftarrow$max(v, MIN-VALUE $(succ)$)
  $v_m \leftarrow$max(v_m, MAX-VALUE $(succ)$)
end for
return max($v, v_m - c$)

(1)

for $succ$ in SUCCESSORS(state) do
  $v \leftarrow$min(v, MAX-VALUE $(succ)$)
end for
return v

(2)

for $succ$ in SUCCESSORS $(state)$ do
  $v \leftarrow$max(v, MIN-VALUE $(succ)$)
end for
return v

(1)

$v_m \leftarrow -\inf$
for $succ$ in SUCCESSORS(state) do
  $temp \leftarrow$ MAX-VALUE $(succ)$
  $v \leftarrow$ min($v, temp$)
  $v_m \leftarrow$max($v_m, temp$)
  $v \leftarrow$max($v_m - c, v$)
end for
return v

(2)

Submit

✔ Correct (7/7 points)