HW3: Written

Friday, March 15, 2019 11:18 AM

CS_188_Fa...

CS 188
Fall 2018

Introduction to
Artificial Intelligence

Written HW 3

Due: Monday 9/17/2018 at 11:59pm (submit via Gradescope)

Policy: Can be solved in groups (acknowledge collaborators) but must be written up individually

Submission: Your submission should be a PDF that matches this template. Each page of the PDF should align with the corresponding page of the template (page 1 has name/collaborators, question 1 begins on page 2, etc.). Do not reorder, split, combine, or add extra pages. The intention is that you print out the template, write on the page in pen/pencil, and then scan or take pictures of the pages to make your submission. You may also fill out this template digitally (e.g. using a tablet.)

First name	
Last name	
SID	
Collaborators	

Q1. One Wish Pacman

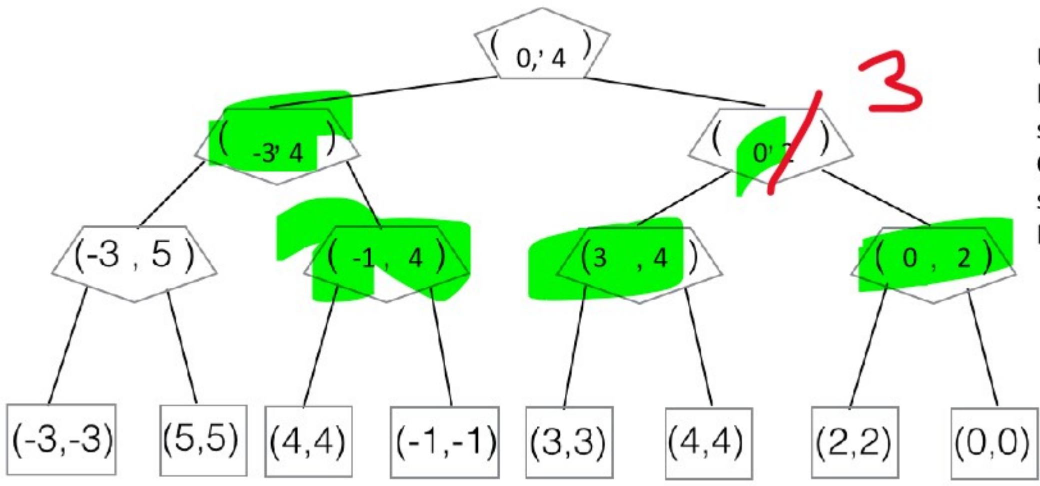
(a) Power Search. Pacman has a special power: *once* in the entire game when a ghost is selecting an action, Pacman can make the ghost choose any desired action instead of the min-action which the ghost would normally take. *The ghosts know about this special power and act accordingly.*

(i) Similar to the minimax algorithm, where the value of each node is determined by the game subtree hanging from that node, we define a value pair (u, v) for each node: u is the value of the subtree if the power is not used in that subtree; v is the value of the subtree if the power is used once in that subtree. For example, in the below subtree with values (-3, 5), if Pacman does not use the power, the ghost acting as a minimizer would choose -3; however, with the special power, Pacman can make the ghost choose the value more desirable to Pacman, in this case 5.

Reminder: Being allowed to use the power once during the game is different from being allowed to use the power in only one node in the game tree below. For example, if Pacman's strategy was to always use the special power on the second ghost then that would only use the power once during execution of the game, but the power would be used in four possible different nodes in the game tree.

For the terminal states we set $u = v = \text{UTILITY}(State)$.

Fill in the (u, v) values in the modified minimax tree below. Pacman is the root and there are two ghosts.



Use a wish
I can take the least score of successors using wish
OR I can take the MAX score of successors without using wish.
Max(min(2, 4) , max(3, 0)) = 3

(ii) Complete the algorithm below, which is a modification of the minimax algorithm, to work in the general case: Pacman can use the power at most once in the game but Pacman and ghosts can have multiple turns in the game.

function VALUE($state$)
  if $state$ is leaf **then**
    $u \leftarrow$ UTILITY($state$)
    $v \leftarrow$ UTILITY($state$)
    return (u, v)
  end if
  if $state$ is Max-Node **then**
    return MAX-VALUE($state$)
  else
    return MIN-VALUE($state$)
  end if
end function

function MAX-VALUE($state$)
  $uList \leftarrow [\,], vList \leftarrow [\,]$
  for $successor$ in SUCCESSORS($state$) **do**
    $(u', v') \leftarrow$ VALUE($successor$)
    $uList$.append(u')
    $vList$.append(v')
  end for
  $u \leftarrow \max(uList)$
  $v \leftarrow \max(vList)$
  return (u, v)
end function

function MIN-VALUE($state$)
  $uList \leftarrow [\,], vList \leftarrow [\,]$
  for $successor$ in SUCCESSORS($state$) **do**
    $(u', v') \leftarrow$ VALUE($successor$)
    $uList$.append(u')
    $vList$.append(v')
  end for

  $u \leftarrow$ Min(uList)

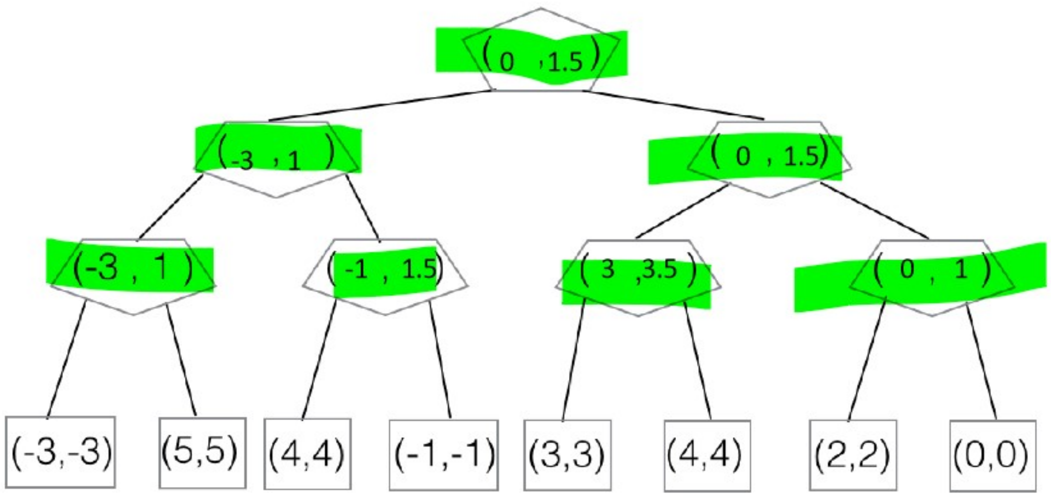
  $v \leftarrow$ Min(max(uList), min(vList))

  return (u, v)
end function

(b) Weak-Power Search. Now, rather than giving Pacman control over a ghost move once in the game, the special power allows Pacman to once make a ghost act randomly. The ghosts know about Pacman's power and act accordingly.

(i) The propagated values (u, v) are defined similarly as in the preceding question: u is the value of the subtree if the power is not used in that subtree; v is the value of the subtree if the power is used once in that subtree.

Fill in the (u, v) values in the modified minimax tree below, where there are two ghosts.



(ii) Complete the algorithm below, which is a modification of the minimax algorithm, to work in the general case: Pacman can use the weak power at most once in the game but Pacman and ghosts can have multiple turns in the game.

Hint: you can make use of a min, max, and average function

```
function VALUE(state)
    if state is leaf then
        u ← UTILITY(state)
        v ← UTILITY(state)
        return (u, v)
    end if
    if state is Max-Node then
        return MAX-VALUE(state)
    else
        return MIN-VALUE(state)
    end if
end function

function MAX-VALUE(state)
    uList ← [ ], vList ← [ ]
    for successor in SUCCESSORS(state) do
        (u', v') ← VALUE(successor)
        uList.append(u')
        vList.append(v')
    end for
    u ← max(uList)
    v ← max(vList)
    return (u, v)
end function

function MIN-VALUE(state)
    uList ← [ ], vList ← [ ]
    for successor in SUCCESSORS(state) do
        (u', v') ← VALUE(successor)
        uList.append(u')
        vList.append(v')
    end for

    u ← Min(uList)

    v ← Min( min(vList), average(uList))

    return (u, v)
end function
```

Q2. Lotteries in Ghost Kingdom

(a) **Diverse Utilities.** Ghost-King (GK) was once great friends with Pacman (P) because he observed that Pacman and he shared the same preference order among all possible event outcomes. Ghost-King, therefore, assumed that he and Pacman shared the same utility function. However, he soon started realizing that he and Pacman had a different preference order when it came to lotteries and, alas, this was the end of their friendship.

Let Ghost-King and Pacman's utility functions be denoted by U_{GK} and U_P respectively. Assume both U_{GK} and U_P are guaranteed to output non-negative values.

(i) Which of the following relations between U_{GK} and U_P are consistent with Ghost King's observation that U_{GK} and U_P agree, with respect to all event outcomes but not all lotteries?

- ☐ $U_P = aU_{GK} + b \quad (0 < a < 1, b > 0)$
- ☐ $U_P = aU_{GK} + b \quad (a > 1, b > 0)$
- ☐ $U_P = U_{GK}^2$
- ☐ $U_P = \sqrt{(U_{GK})}$

Ugk1 > Ugk2 implies Up1 > Up2

(ii) In addition to the above, Ghost-King also realized that Pacman was more risk-taking than him. Which of the relations between U_{GK} and U_P are possible?

- ☐ $U_P = aU_{GK} + b \quad (0 < a < 1, b > 0)$
- ☐ $U_P = aU_{GK} + b \quad (a > 1, b > 0)$
- ☐ $U_P = U_{GK}^2$
- ☐ $U_P = \sqrt{(U_{GK})}$

Up will get quadratic increase in benefit as Ugk utility increases

(b) **Guaranteed Return.** Pacman often enters lotteries in the Ghost Kingdom. A particular Ghost vendor offers a lottery (for free) with three possible outcomes that are each equally likely: winning \$1, \$4, or \$5.

Let $U_P(m)$ denote Pacman's utility function for $\$m$. Assume that Pacman always acts rationally.

(i) The vendor offers Pacman a special deal - if Pacman pays \$1, the vendor will manipulate the lottery such that Pacman **always gets the highest reward possible**. For which of these utility functions would Pacman choose to pay the \$1 to the vendor for the manipulated lottery over the original lottery? (Note that if Pacman pays \$1 and wins $\$m$ in the lottery, his actual winnings are $\$m$-1.)

- ☐ $U_P(m) = m$
- ☐ $U_P(m) = m^2$

No pay: m = 10/3. Up(m) = 10/3. Up(m) = 100/9
Pay: m = 4. Up(m) = 4. Up(m) = 16

(ii) Now assume that the ghost vendor can only manipulate the lottery such that Pacman **never gets the lowest reward** and the remaining two outcomes become equally likely. For which of these utility functions would Pacman choose to pay the \$1 to the vendor for the manipulated lottery over the original lottery?

- ☐ $U_P(m) = m$
- ☐ $U_P(m) = m^2$

	Pay	NoPay
M	3.5	3.333
M^2	12.25	14

100/9 = 11.1111

3.5*3.5 = 12.25

=16+25 =41
(41)/2-1=19.5

1+16+25 = 42
42 / 3 = 14