

Self-assessment due: Monday 9/24/2018 at 11:59pm (submit via Gradescope)

Instructions for self-assessment: Take your original submission and annotate any differences from the provided solutions. For **each subpart** where your original answer was correct, **write “correct”** to demonstrate that you have checked your work. For each subpart where your original answer was incorrect, write out the correct answer and comment on the difference between your answer and the explanation provided in the solutions. You should complete your self-assessment using a **different color** of ink from your original work. If you need to, you can download a PDF copy of your submission from Gradescope.

Your submission must be a 5-page PDF that follows the template (page 1 has name/collaborators, question 1 begins on page 2, etc.). Do not reorder, split, combine, or add extra pages. If your original homework submission did not follow the correct format, **you must fix the format to receive credit on your self-assessment.**

If you did not complete some questions in your original submission, first complete those questions without consulting the solutions and then use a different color of ink to conduct a self-assessment.

Q1. One Wish Pacman

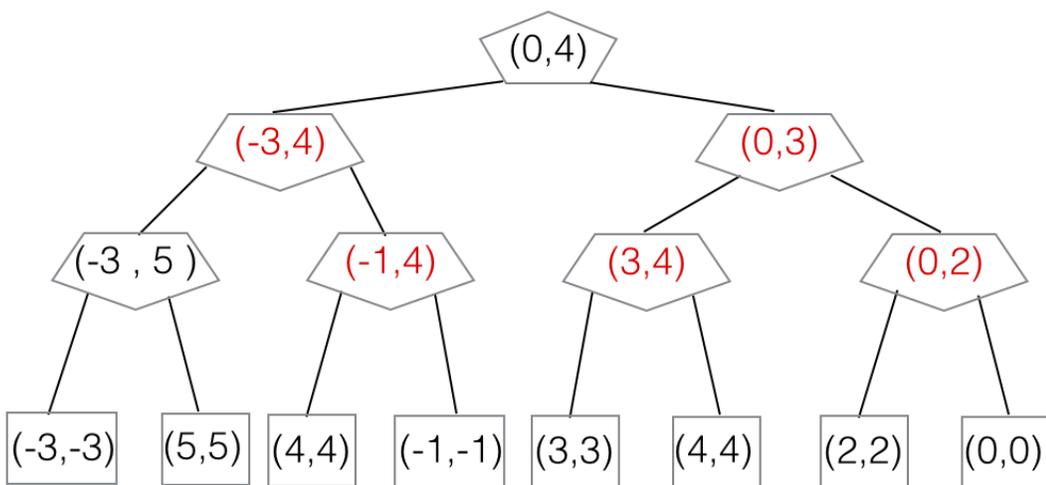
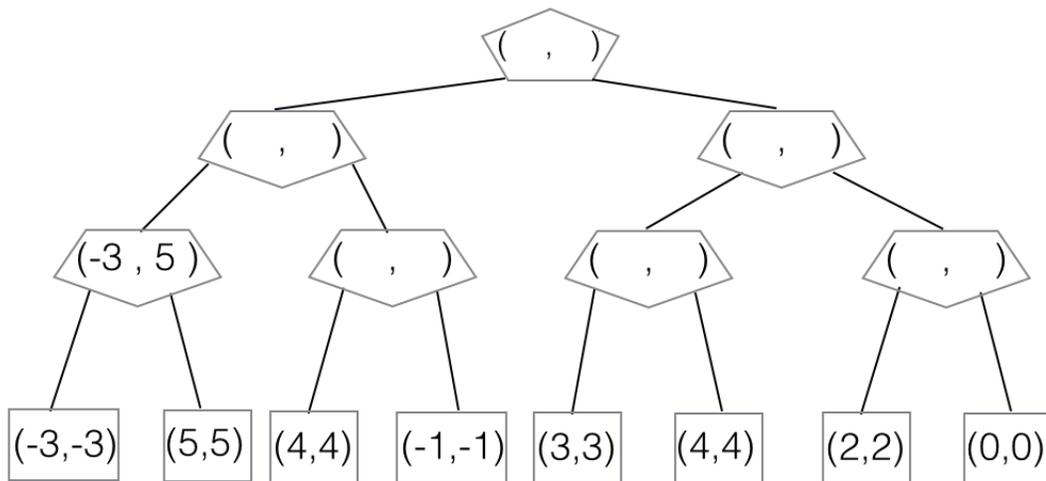
(a) **Power Search.** Pacman has a special power: *once* in the entire game when a ghost is selecting an action, Pacman can make the ghost choose any desired action instead of the min-action which the ghost would normally take. The ghosts know about this special power and act accordingly.

(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 $u = v = \text{UTILITY}(\text{State})$.

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



Please see the solution of the general algorithm in the next part to see how the u, v values get propagated up the game tree.

- (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
     $u \leftarrow \text{UTILITY}(\textit{state})$ 
     $v \leftarrow \text{UTILITY}(\textit{state})$ 
    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
    ( $u', v'$ )  $\leftarrow \text{VALUE}(\textit{successor})$ 
     $uList.append(u')$ 
     $vList.append(v')$ 
  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
    ( $u', v'$ )  $\leftarrow \text{VALUE}(\textit{successor})$ 
     $uList.append(u')$ 
     $vList.append(v')$ 
  end for

```

$u \leftarrow \underline{\min(uList)}$

$v \leftarrow \underline{\max(\max(uList), \min(vList))}$

```

  return ( $u, v$ )
end function

```

The u value of a min-node corresponds to the case if Pacman does not use his power in the game subtree hanging from the current min-node. Therefore, it is equal to the minimum of the u values of the children of the node.

The v value of the min-node corresponds to the case when pacman uses his power once in the subtree. Pacman has two choices here - a) To use the power on the current node, or b) To use the power further down in the subtree.

In case a), the value of the node corresponds to choosing the best among the children's u values = $\max(uList)$ (we consider u values of children as Pacman is using his power on this node and therefore, cannot use it in the subtrees of the node's children).

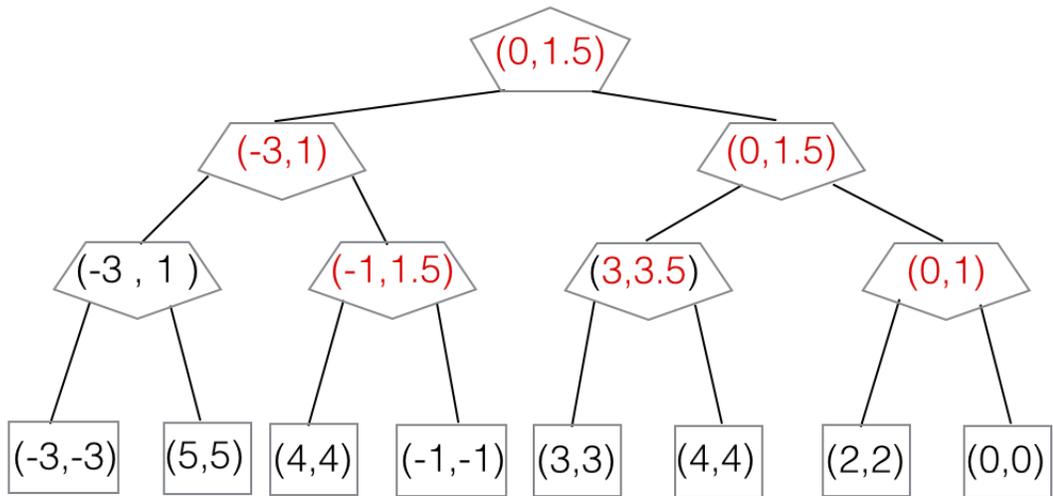
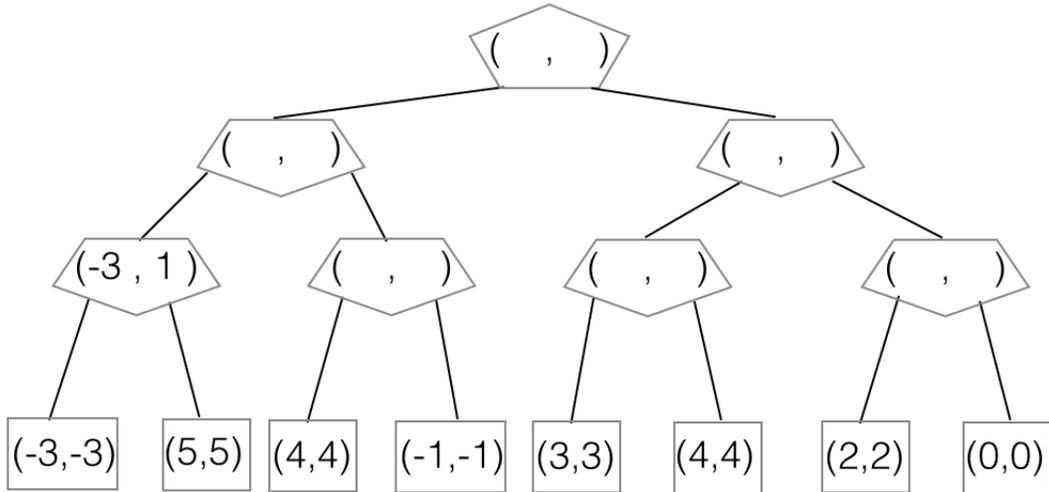
In case b), Pacman uses his power in one of the child subtrees so we consider the v values of the children, Since Pacman is not using his power on this node, the current node acts as a minimizer, making the value in case b) = $\min(vList)$

The v value at the current node is the best of the above two cases.

(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.



Please see the solution of the general algorithm in the next part to see how the u, v values get propagated up the game tree.

(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
     $u \leftarrow \text{UTILITY}(\textit{state})$ 
     $v \leftarrow \text{UTILITY}(\textit{state})$ 
    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

```

$u \leftarrow \underline{\min(uList)}$

$v \leftarrow \underline{\max(\text{avg}(uList), \min(vList))}$

```

  return ( $u, v$ )
end function

```

The solution to this scenario is same as before, except that when considering case a) for the v value of a min-node, the value of the node corresponds to choosing the average of the children's u values = $\text{avg}(uList)$

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?

$U_P = aU_{GK} + b \quad (0 < a < 1, b > 0)$

$U_P = aU_{GK} + b \quad (a > 1, b > 0)$

$U_P = U_{GK}^2$

$U_P = \sqrt{U_{GK}}$

For all the above options, U_P and U_{GK} result in the same preference order between two non-lottery events ($U_P(e_1) > U_P(e_2) \Leftrightarrow U_{GK}(e_1) > U_{GK}(e_2)$). However, options 1 and 2 also share the same preference order among all lotteries as well.

- (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?

$U_P = aU_{GK} + b \quad (0 < a < 1, b > 0)$

$U_P = aU_{GK} + b \quad (a > 1, b > 0)$

$U_P = U_{GK}^2$

$U_P = \sqrt{U_{GK}}$

As an example, say Ghost-King prefers winning \$2 as much as a lottery : winning \$0 or \$4 with equal probability. For option c), Pacman prefers the lottery much more (more risk-taking) and for option d), Pacman prefers the guaranteed reward.

- (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 \$ $m-1$.)

$U_P(m) = m$

$U_P(m) = m^2$

a) $U_P(m) = m$:

If pacman does not pay, expected utility = $\frac{1}{3}(5) + \frac{1}{3}(4) + \frac{1}{3}(1) = \frac{10}{3}$

If pacman pays up, expected utility = $1(5-1) + 0(4-1) + 0(1-1) = 4$.

b) $U_P(m) = m^2$:

If pacman does not pay, expected utility = $\frac{1}{3}(5)^2 + \frac{1}{3}(4)^2 + \frac{1}{3}(1)^2 = 14$

If pacman pays up, expected utility = $1(5-1)^2 + 0(4-1)^2 + 0(1-1)^2 = 16$.

- (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?

$U_P(m) = m$

$U_P(m) = m^2$

a) $U_P(m) = m$:

If pacman does not pay, expected utility = $\frac{1}{3}(5) + \frac{1}{3}(4) + \frac{1}{3}(1) = \frac{10}{3}$

If pacman pays up, expected utility = $\frac{1}{2}(5 - 1) + \frac{1}{2}(4 - 1) + 0(1 - 1) = 3.5$.

b) $U_P(m) = m^2$:

If pacman does not pay, expected utility = $\frac{1}{3}(5)^2 + \frac{1}{3}(4)^2 + \frac{1}{3}(1)^2 = 14$

If pacman pays up, expected utility = $\frac{1}{2}(5 - 1)^2 + \frac{1}{2}(4 - 1)^2 + 0(1 - 1)^2 = 12.5$.