

CSCE 580, Fall 2020

Assignment 3

Username: primiani@email.sc.edu

Name: Jack Primiani

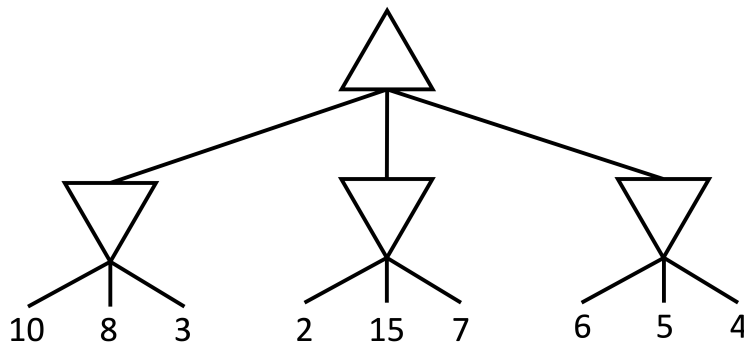
By turning in this assignment, I agree by the honor code of USC Columbia.

Submission. You need to submit the following files to Blackboard:

- A pdf file named as assignment3- $\langle username \rangle$.pdf, where you replace $\langle username \rangle$ with your email username. This pdf file contains your answers to Problems 1 and 2: Edit the assignment3.tex file to fill in your answers for Problems 1 and 2, and submit the pdf file generated from the edited tex file.
- A zip file named as assignment3- $\langle username \rangle$.zip, where you replace $\langle username \rangle$ with your email username. This zip file contains the single file of multiAgents.py specified in Problem 3.
- A html file named as assignment3- $\langle username \rangle$.html, where you replace $\langle username \rangle$ with your email username. The html file contains the autograder's results. Please see Problem 3 for details.

Problem 1 [10 pt]: Two-player zero-sum games

Consider the zero-sum game tree shown below. Triangles that point up, such as at the top node (root), represent choices for the maximizing player; triangles that point down represent choices for the minimizing player.



(a) What are the minimax values of each of the four nodes?

P, Q, R are min-nodes:

$$U(P) = \min 10, 8, 3$$

$$U(Q) = \min 2, 15, 7$$

$$U(R) = \min 6, 5, 4$$

S is a max-node:

$$U(S)=3,2,4 = 4$$

- (b) Which leaves can be pruned from the game tree above through alpha-beta pruning? If no nodes can be pruned, explain why not. Assume the search goes from left to right: when choosing which child to visit first, choose the left-most unvisited child.

No nodes can be pruned. There will always be the possibility that some leaf further down the branch will have a very high value, which increases the overall average value.

- (c) Again, consider the same game tree, except that now, instead of a minimizing player, we have a chance node that will select one of the three values uniformly at random. What are the expectimax value of each of the four nodes?

P, Q, R are children of S:

$$\text{Expectimax of P} = 7$$

$$\text{Expectimax of Q} = 8$$

$$\text{Expectimax of R} = 5$$

$$\text{Expectimax of S} = 8$$

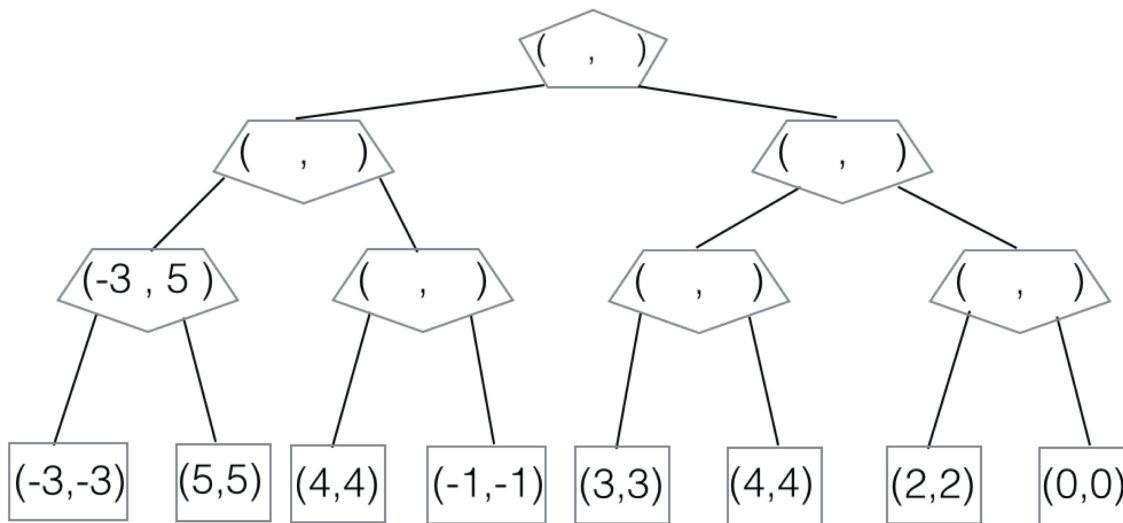
Problem 2 [20 pt]: One Wish Pacman

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.*

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 $u = v = \text{Utility}(\text{state})$.



- (a) Find the (u, v) values in the modified minimax tree above. Pacman is the root and there are two ghosts.

$(0, 4)$

$(-3, 4)$ $(0, 3)$

$(-3, 5)$ $(-1, 4)$ $(3, 4)$ $(0, 2)$

- (b) 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. Briefly explain your completion of the two blanks.

```

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

```

$u \leftarrow$ _____ (i)

$v \leftarrow$ _____ (ii)

```

  return ( $u, v$ )
end function

```

$u \leftarrow \min(uList)$

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

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

The v value is when pacman uses his power once in the subtree. Pacman only has two options to use the power on the current node or to use the power further down in the subtree.

Problem 3 [70 pt]: Multi-agent Pacman

This is a programming problem adapted from CSCE 580 in Spring 2019. Please use the following link for the problem specification: <https://pooyanjamshidi.github.io/csce580/project2/>.

You need to submit 1) a .zip file containing the multiAgents.py file you edited, and 2) a .html file generated by the following command: `python autograder.py --edx-output`

The .html file contains the autograder's results.