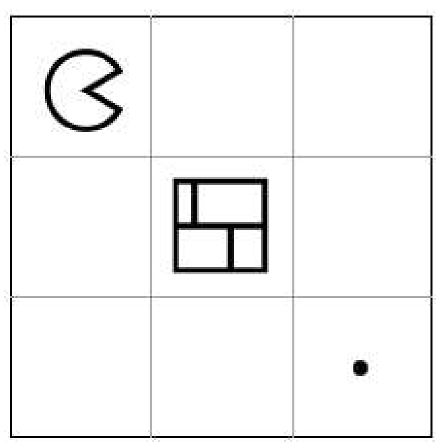
Q4: Surrealist Pacman

Problem 4: Surrealist Pacman

In the game of Surrealist Pacman, Pacman plays against a moving wall. On Pacman's turn, Pacman must move in one of the four cardinal directions, and must move into an unoccupied square. On the wall's turn, the wall must move in one of the four cardinal directions, and must move into an unoccupied square. The wall cannot move into a dot-containing square. Staying still is not allowed by either player. Pacman's score is always equal to the number of dots he has eaten.

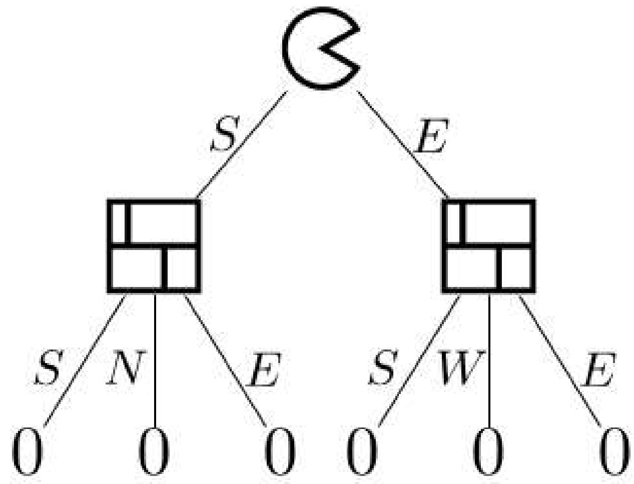
The first game begins in the configuration shown below. Pacman moves first.

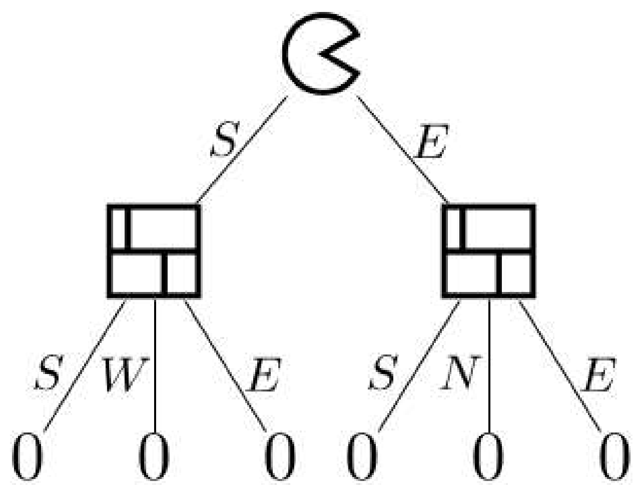


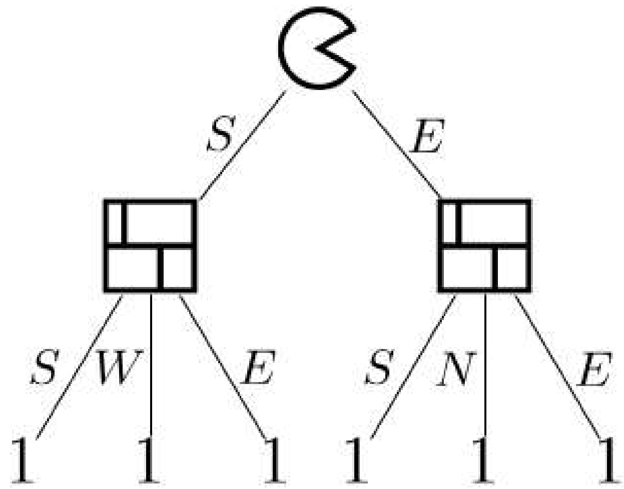
Part 1

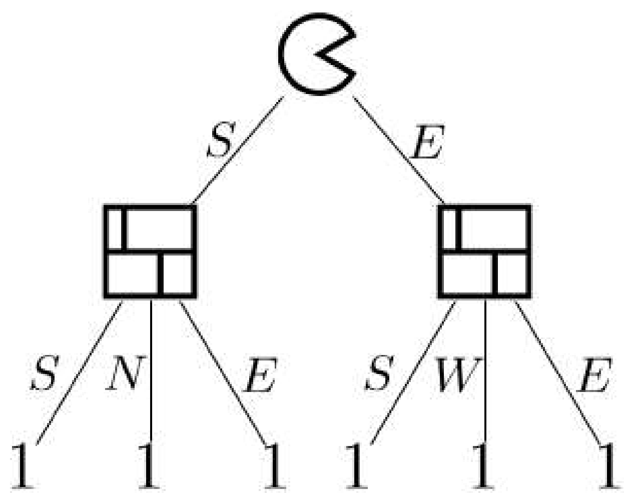
1/1 point (ungraded)

Which of the following is the game tree with one move for each player? Nodes in the tree represent game states (location of all agents and walls). Edges in the tree connect successor states to their parent states.


S
E
S
N
E
S
W
E
0
0
0
0
0
0


S
E
S
W
E
S
N
E
0
0
0
0
0
0





Submit

✔ Correct (1/1 point)

Part 2

0.0/1.0 point (ungraded)

According to the depth-limited game tree you selected above, what is the value of the game? Use Pacman's score as your evaluation function.

0

Submit

Part 3

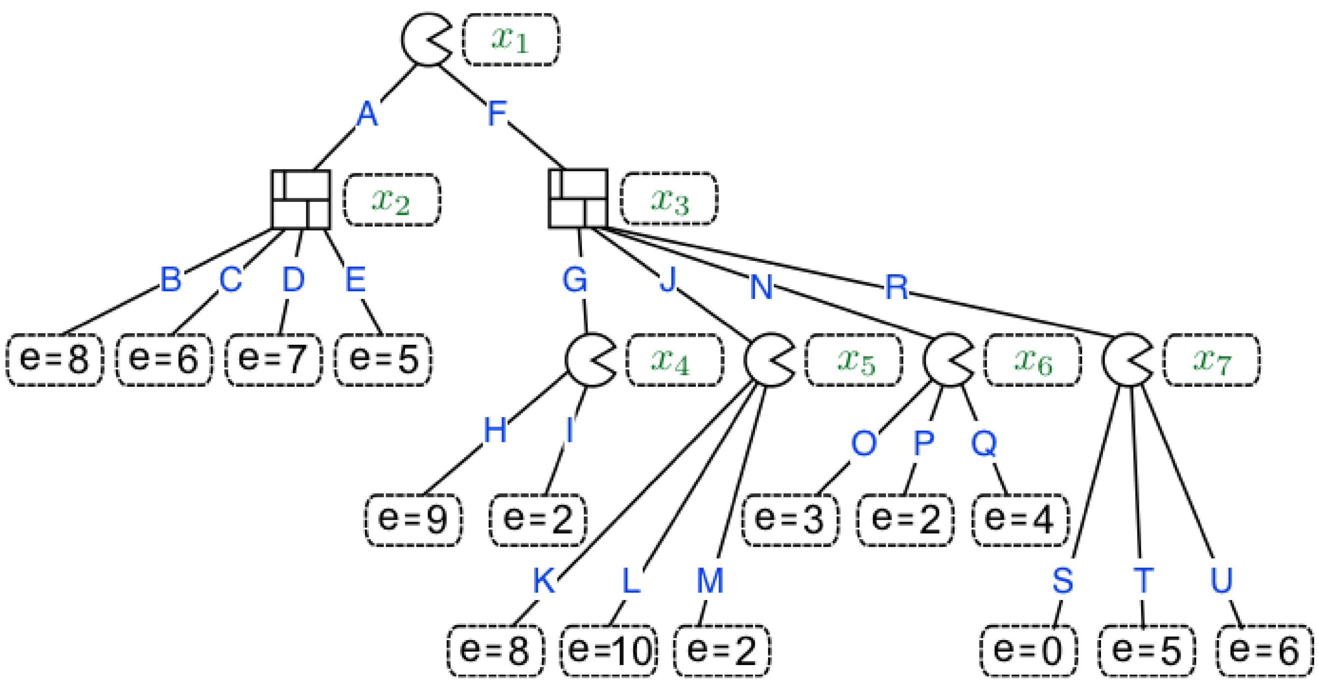
0.0/2.0 points (ungraded)

If we were to consider a game tree with ten moves for each player (rather than just one), what would be the value of the game as computed by minimax?

1

Submit

A second game is played on a more complicated board. A partial game tree is drawn, and leaf nodes have been scored using an (unknown) evaluation function *e*.



Fill in each of the following values for the internal nodes of the game tree above using the minimax algorithm.

Part 4

1/1 point (ungraded)

$x_1 =$

5

Submit

Correct (1/1 point)

Part 5

1/1 point (ungraded)

$x_2 =$

5

Submit

Correct (1/1 point)

Part 6

1/1 point (ungraded)

$x_3 =$

4

Submit

Correct (1/1 point)

Part 7

1/1 point (ungraded)

$x_4 =$

9

Submit

Correct (1/1 point)

Part 8

1/1 point (ungraded)

$x_5 =$

10

Submit

Correct (1/1 point)

Part 9

1/1 point (ungraded)

$x_6 =$

4

Submit

Correct (1/1 point)

Part 10

1/1 point (ungraded)

$x_7 =$

6

Submit

Correct (1/1 point)

Part 11

3/3 points (ungraded)

Mark any branches that are not evaluated when using alpha-beta pruning (assuming the standard left-to-right traversal of the tree). If the pruning happens high up in the tree, only mark the branch at which the pruning actually happens, for example if branch A is pruned, don't mark B, C, D, or E.

- [ ] A
- [ ] B
- [ ] C
- [ ] D
- [ ] E
- [ ] F
- [ ] G
- [ ] H

- [ ] I
- [ ] J
- [ ] K
- [ ] L
- [x] M
- [ ] N
- [ ] O
- [ ] P
- [ ] Q
- [x] R
- [ ] S
- [ ] T
- [ ] U

✔

Submit

✔ Correct (3/3 points)

Suppose that this evaluation function has a special property: it is known to give the correct minimax value of any internal node to within 2, and the correct minimax values of the leaf nodes exactly. That is, if v is the true minimax value of a particular node, and e is the value of the evaluation function applied to that node, $e - 2 \leq v \leq e + 2$, and $v = e$ if the node is a dashed box in the tree below.

Using this special property, you can modify the alpha-beta pruning algorithm to prune more nodes.

Part 12

2/2 points (ungraded)

Standard alpha-beta pseudocode is given below (only the max-value recursion). Select the choices below to replace the corresponding boxes in the pseudocode shown so that the pseudocode prunes as many nodes as possible, taking into account this special property of the evaluation function.

```
function MAX-VALUE(node, α, β)
    e ← EVALUATIONFUNCTION(node)
    if node is leaf then
        return e
    end if
    [          ] (1)
    v ← −∞
    for child ← CHILDREN(node) do
        [v ← MAX(v, MIN-VALUE(child, α, β))] (2)
        if v ≥ β then
            return v
        end if
        α ← MAX(α, v)
    end for
    return v
end function
```

- (1):
  ```
  if e − 2 ≥ α then
    return e − 2
  end if
  ```
  (2): $v \leftarrow \text{MAX}(v, \text{MIN}-\text{VALUE}(child, \text{MAX}(\alpha, e-2), \text{MIN}(\beta, e+2)))$

- (1):
  ```
  if e + 2 ≤ β then
    return e − 2
  end if
  ```
  (2): $v \leftarrow \text{MAX}(v, \text{MIN}-\text{VALUE}(child, \text{MAX}(\alpha, e+2), \text{MIN}(\beta, e-2)))$

- [x] (1)
```
if e − 2 ≥ β then
   return e − 2
end if
```
(2) $v \leftarrow \text{MAX}(v, \text{MIN}-\text{VALUE}(child, \text{MAX}(\alpha, e-2), \text{MIN}(\beta, e+2)))$

- [ ] (1)
```
if e + 2 ≤ α then
   return e + 2
end if
```
(2) $v \leftarrow \text{MAX}(v, \text{MIN}-\text{VALUE}(child, \text{MAX}(\alpha, e+2), \text{MIN}(\beta, e-2)))$

Submit

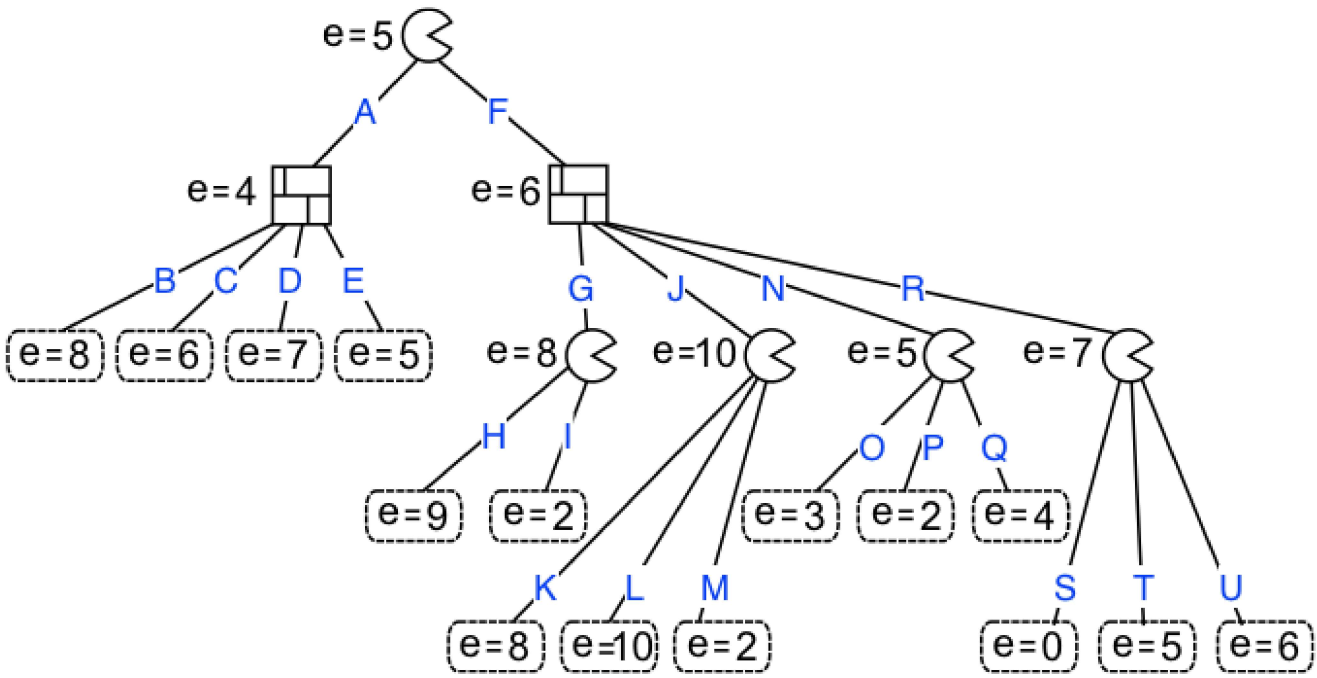
Correct (2/2 points)

Part 13

3/3 points (ungraded)

The same game tree is shown below, with the evaluation function applied to *internal* as well as leaf nodes.

In the game tree below cross off any nodes that can be pruned assuming the special property holds true. If the pruning happens high up in the tree, only mark the branch at which the pruning actually happens, for example if branch A is pruned, don't mark B, C, D, or E.



- [ ] A
- [ ] B
- [ ] C
- [ ] D
- [ ] E
- [ ] F
- [ ] G
- [ ] H
- [x] I
- [ ] J

- [x] K
- [x] L
- [x] M
- [ ] N
- [ ] O
- [ ] P
- [ ] Q
- [x] R
- [ ] S
- [ ] T
- [ ] U

Submit

Correct (3/3 points)