Week 5

14 試題

1 point

1 。
Which of the following statements is true?

- In a fully observable, turn-taking, zero-sum game between two perfectly rational players, it does not help the first player to know what strategy the second player is using, that is, what move the second player will make, given the first player's move.
- A perfectly rational agent is omniscient
- In a partially observable, turn-taking, zero-sum game between two perfectly rational players, it does not help the first player to know what move the second player will make, given the first player's move.
- A perfectly rational agent never loses.

1 point

2 。
There are two 8-puzzles, and two players take turns moving. A fair coin is flipped to determine the puzzle on which to make a move at that turn. The winner is the first to solve one puzzle.

(a) What kind of search can we apply?

- MiniMax search
- LRTA*
- RBFS
- ExpectiMiniMax

1 point

3 。
(b) Corresponding to the above question,

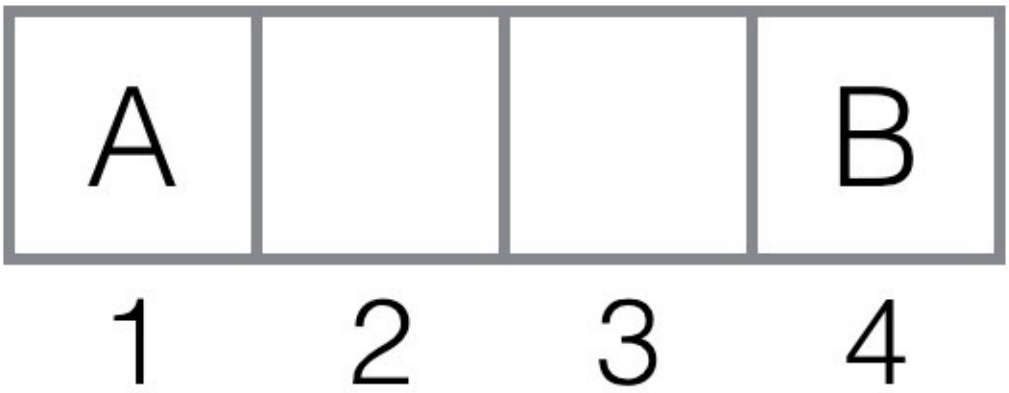
can we apply NegaScout?

If yes, apply on what kind of node?

- No
- Yes, on chance node
- Yes, on Min node

1 point

4 。
Consider a game:



In the beginning, player A with his token “A” is at position 1, and player B with his token “B” at position 4. Player A moves first. Two players take turns to move, and each player must move his token to an adjacent space in neither direction. If the opponent is in an adjacent space, the player may jump over the opponent to the next open space(ex: in player B’s turn, A is at position 2 and B is at position 3, B can jump over A to position 1). The game ends when one player reaches the opposite end of the board. If player A reached position 4, then the value of the game to A is +1; if B reached position 1, the value of the game to A is -1. Can we apply MiniMax tree search? Why or why not?

- No, MiniMax may fail due to infinite loop search
- Yes, we can use NegaScout to prun the infinite loop

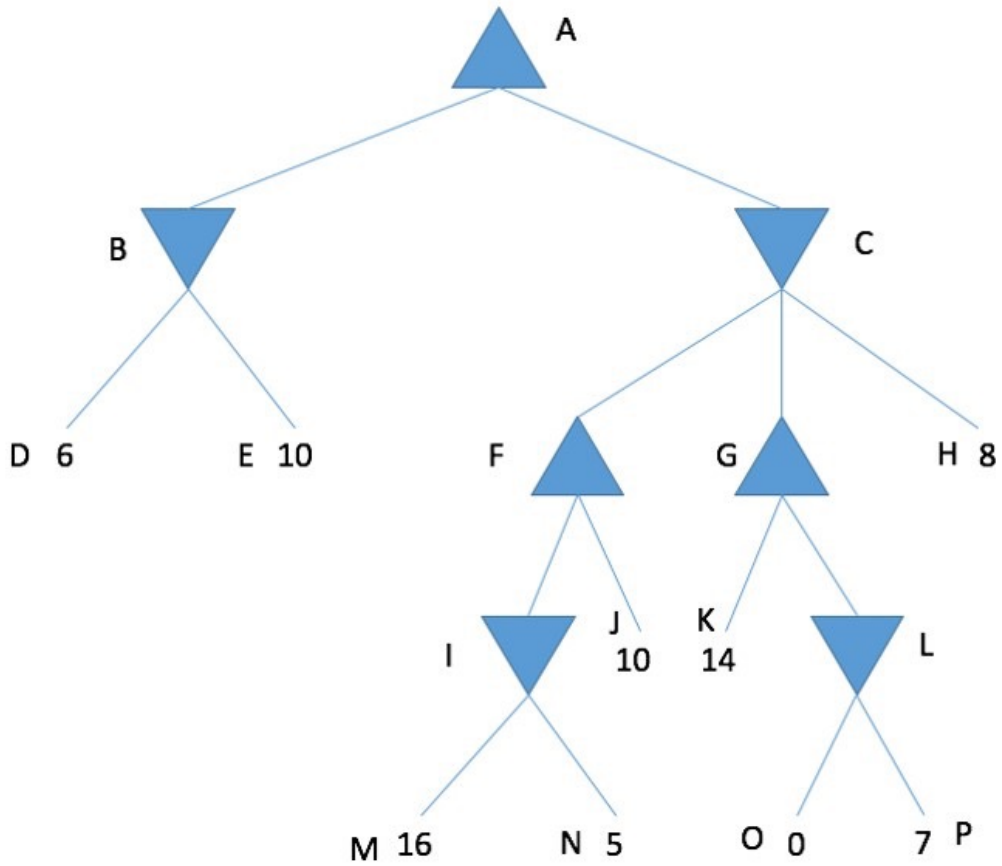
- No, we can not formulate this problem to apply MiniMax
- Yes, the first move player wins.

---

1 point

5 。

Consider the following MiniMax tree



(a) Which nodes will be pruned if we apply the left-to-right α − β pruning?

What about right-to left?

- Left-to-right: H; right-to-left: I, M, N
- Left-to-right: H; right-to-left: M
- Left-to-right: P; right-to-left: D, M
- Left-to-right: P; right-to-left: I, M, N

---

1 point

6 ◦

(b) Corresponding to the above question,

if we apply NegaScout from left to right, which node(s) will be pruned?

Do we need to full re-search?

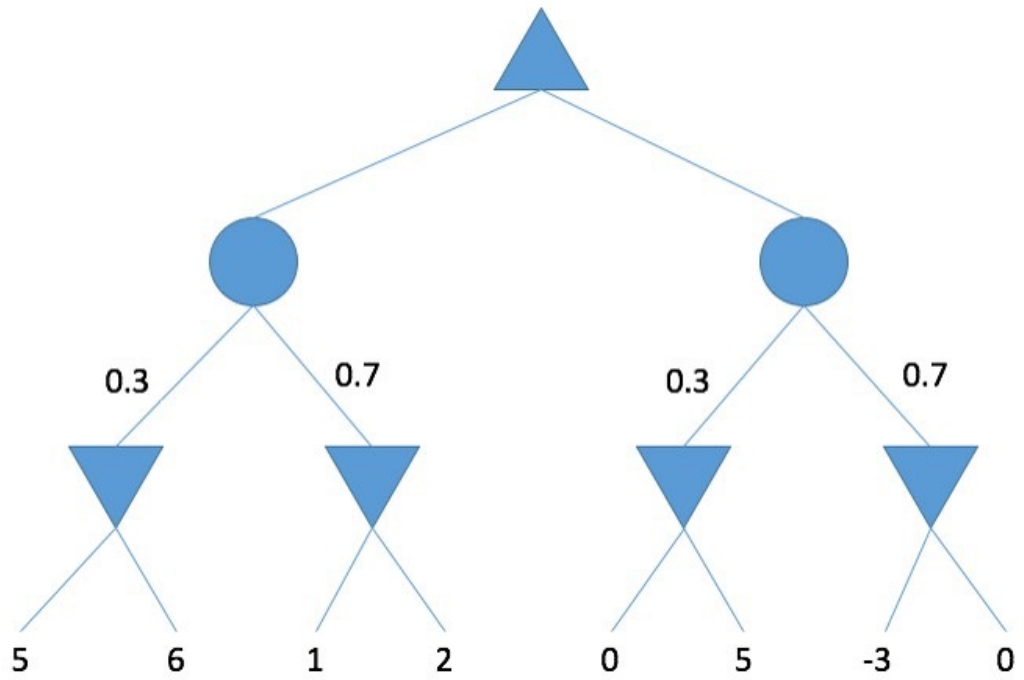
- N, H; Yes
- L, O, P; No
- L, O, P; Yes
- N, H; No

---

1 point

7 ◦

Given MiniMax tree with chance nodes, chance nodes are circle, and the number show beside the branch is the probability



We perform a left-to-right ExpectiMiniMax search.

(a) Which is the best branch?

- Left

- Right

---

1 point

8 ◦
(b) Corresponding to the above question,

if the last two leaves are unknown, can we decide which branch to go?

How about only last leaf?

- Yes; No
- No; Yes
- Yes; Yes
- No; No

---

1 point

9 ◦
In the following, a “min” tree consists only of min nodes, whereas an “expectimin” tree consists of a max at the root with alternating layers of chance and min nodes. At chance nodes, all outcome probabilities are nonzero.

(a) Assuming the depth is finite, but leaf values are unbounded, is pruning ever possible in a min tree?

- No
- Yes

---

1 point

10 ◦
(b) Corresponding to the above question,

is pruning ever possible in an expectimin tree with same condition as (a)?

- No

- Yes

1 point

11 ◦

(c) Corresponding to the above question,

if leaf values are all > 0, is pruning ever possible in min tree?

- No
- Yes

1 point

12 ◦

(d) Corresponding to the above question,

if leaf values are all > 0, is pruning ever possible in expectimin tree?

- No
- Yes

1 point

13 ◦

(e) Corresponding to the above question,

if leaf values are all in range [0, 1], is pruning ever possible in min tree?

- No
- Yes

1 point

14 ◦