Instructions:

- *Your solutions for this tutorial must be TYPE-WRITTEN.*
- *Submit your solution(s) to your tutor. You can make another copy for yourself if necessary. Late submission will NOT be entertained.*
- *YOUR SOLUTION TO QUESTION* 1 *will be GRADED for this tutorial.*
- *You can work in pairs, but each of you should submit the solution(s) individually.*
- *Include the name of your collaborator in your submission.*

1. Prove that α-β pruning does not remove any strategies that are played in a Nash equilibrium of an extensive form game; can α-β pruning be used to find a subgame-perfect Nash equilibrium? Prove or provide a counterexample.

   Solution: Suppose α-β pruning removes a subtree, say a subtree rooted in a node u_1 that's the child of a node v. Suppose that v is a MAX node; then there is some other node u_2 who's a child of v and the payoff to the MAX player from choosing u_2 is at least as high as that of choosing u_1 (that's the only reason we'll prune u_1 in the first place). In other words, let x be the payoff that player 1 gets from choosing the node u_1 from v; then it must be the case that x is no more than the payoff to player 1 from playing u_2.

   Therefore, pruning u_1 will never prune a strategy that's played in a Nash equilibrium of the game overall. However, since we do not explore the entire tree, we do not specify a complete strategy, and in particular, we cannot be specifying a sub-perfect Nash equilibrium strategy.

2. Consider the following game: we have an attacker looking at three targets: t_1, t_2 and t_3. A defender must choose which of the two targets it will guard; however, the attacker has an advantage: it can observe what the defender is doing before it chooses its move. If an attacker successfully attacks it receives a payoff of 1 and the defender gets a payoff of -1.

   (a) Model this problem as a minimax search problem. Draw out the search tree. What is the defender's payoff in this game?

   (b) Can the defender do better by randomizing? What is the defender's optimal strategy? Prove your claim.

   Solution:

   (a) The search tree is given in Figure 1. Note that the attacker can choose an action that guarantees it a payoff of 1 no matter what the defender does.

   (b) However, when the defender randomizes, the attacker's benefit becomes significantly smaller. By choosing the target not to defend with probability $\frac{1}{3}$ its expected loss becomes $\frac{1}{3}$. Note that when the number of targets is n, this probability is $\frac{1}{n}$.

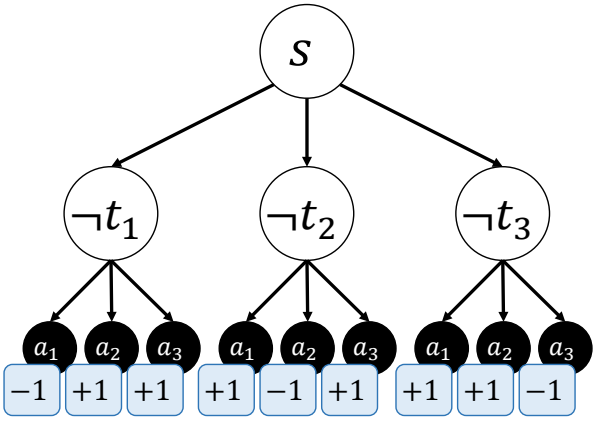


Figure 1: Defender/Attacker game tree. An action by the defender is which node it is *not* defending, given by $\neg t_i$.

3. Assume that we have the following initial state and goal state for the 8-puzzle game. We will use h_1 defined as "the number of misplaced tiles" to evaluate each state.

1	2	8
	4	3
7	6	5

initial state

1	2	3
8		4
7	6	5

goal state

   (a) Apply the hill-climbing search algorithm in Figure 4.2 of AIMA 3rd edition. Can the algorithm reach the goal state?

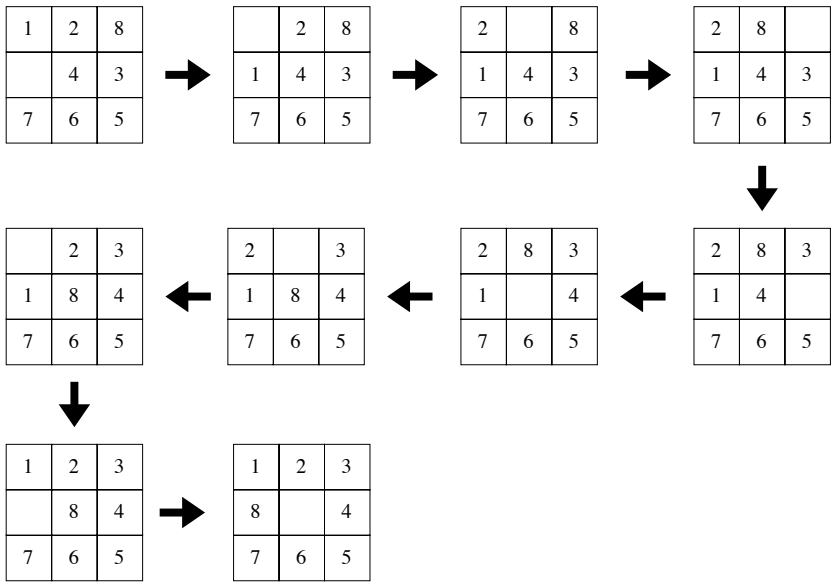
   (b) Identify a sequence of actions leading from the initial state to the goal state.

Solution:

   (a) According to Figure 4.2 of AIMA 3rd edition, the hill-climbing search algorithm will move to a neighboring state only if the neighboring state is better. Since no successor

of the initial state has a better h_1 value, the algorithm is stuck in a local optimum and will not reach the goal state.

(b) The following sequence of actions leads from the initial state to the goal state:



4. Consider the two-player game described in Figure 2. For example, if A is on 3 and B is on 2, then A may move back to 1. The game ends when one player reaches the opposite end of the board. If player A reaches space 4 first, then the value of the game to A is $+1$, if player B reaches space 1 first, then the value of the game to A is -1.

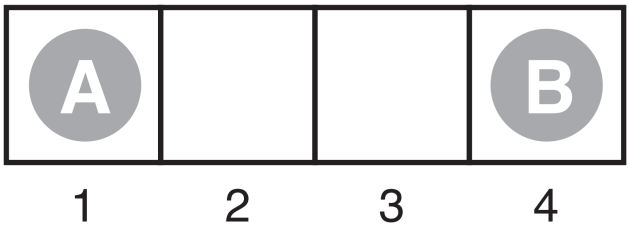


Figure 2: The starting position of a simple game. Player A moves first. The two players take turns moving, and each player must move his token to an open adjacent space in either direction. If the opponent occupies an adjacent space, then a player may jump over the opponent to the next open space if any.

(a) Draw the complete game tree, using the following conventions:

- Write each state as (s_A, s_B), where s_A and s_B denote the token locations.
- Put each terminal state in a square box and write its game value in a circle.

- Put loop states (states that already appear on the path to the root) in double square boxes. Since their value is unclear, annotate each with a "?" in a circle.

(b) Explain why the standard minimax algorithm would fail on this game tree and briefly sketch how you might fix it, drawing on your answer to (a). Does your modified algorithm give optimal decisions for all games with loops?

(c) This 4-square game can be generalized to n squares for any $n > 2$. Prove that A wins if n is even and loses if n is odd.

Solution:

(a) The game tree, complete with annotations of all minimax values, is shown in Figure 3 The terminal states are in single boxes, loop states in double boxes. Each state is

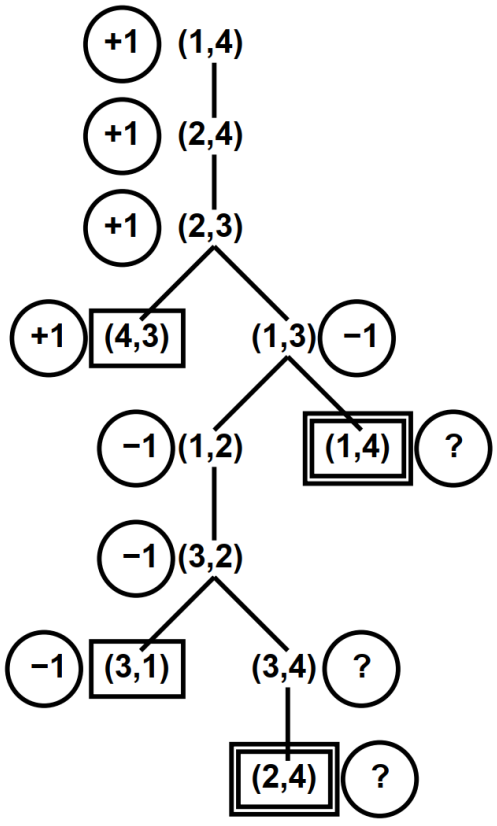


Figure 3: The game tree for the four-square game.

annotated with its minimax value in a circle.

(b) As standard minimax is depth-first and would go into an infinite loop, then the standard minimax algorithm would fail on this game tree. If the game tree has cycles, then a dynamic programming method must be used, as explained in Chapter 17. These algorithms can determine whether each node has a well-determined value (as in this example) or is really an infinite loop in that both players prefer to stay in the loop (or have no choice). In such a case, the rules of the game will need to define the value (otherwise the game will never end). In this game, we can set both values of the loops at $(1, 4)$ and $(2, 4)$ as γ, where $\gamma > -1$. If minimax is still used, it will give optimal decisions.

(c) This question is a little tricky. One approach is a proof by induction on the size of the game. Clearly, the base case $n = 3$ is a loss for A and the base case $n = 4$ is a win for A. For any $n > 4$, the initial moves are the same: A and B both move one step towards each other. Now, we can see that they are engaged in a subgame of size $n - 2$ on the squares $[2, \ldots, n - 1]$, except that there is an extra choice of moves on squares 2 and $n - 1$. Ignoring this for a moment, it is clear that if the "$n - 2$" is won for A, then A gets to the square $n - 1$ before B gets to square 2 (by the definition of winning) and therefore gets to n before B gets to 1, hence the "n" game is won for A. By the same line of reasoning, if "$n - 2$" is won for B then "n" is won for B. Now, the presence of the extra moves complicates the issue, but not too much. First, the player who is slated to win the subgame $[2, \ldots, n - 1]$ never moves back to his home square. If the player slated to lose the subgame does so, then it is easy to show that he is bound to lose the game itself—the other player simply moves forward and a subgame of size $n - 2k$ is played one step closer to the loser's home square.