# Analysis: Stack Management

## Stack Management: Analysis

This problem requires quite a lot of thinking, but relatively little code in the end!

Let S be the number of suits that appear on at least one card used in the problem. For each of these suits, we will call the card with the highest value the *ace* of that suit, and the second-highest card (if it exists) the *king*. We'll say that a card is *visible* if it is at the top of one of the stacks, and that a suit is *visible* if a card of that suit is visible.

Notice that once a suit becomes visible, it will stay visible throughout the rest of the game. At any point in the game, if there are **N** visible suits (recall that **N** is the number of stacks), then either we have won, or we cannot make any more moves and so we have lost. On the other hand, if there are fewer than **N** visible suits, then either we have won, or we are able to make a move (because either a suit has two visible cards, or there is an empty stack). In particular, this implies that if S < **N**, then we are guaranteed to win however we play. However, if S > **N**, we cannot win, because we can never remove the last card of any suit from the game. This means that S = **N** is the only interesting case, so we will assume from now on that S = **N**. In this case, a winning position is a position in which we have exactly one card in each stack, with each one representing a different suit.

Let us assume there is some way to win the game, and we will consider the very last move of that game. Before that move, we had not won, and yet we were able to make a move; this means there must have been fewer than **N** suits visible. So, the last move must have exposed a card of some suit that had never been visible. This means that this suit contained only one card (the card that now remains as the only card in its stack), and this card started the game at the bottom of its stack.

Also note that it is never disadvantageous to remove a card; the only real decisions made in the game are choosing which cards to move into an empty stack. Thus, as a pre-processing step, we can remove all the cards that can be removed from the initial position. If doing that is enough to win the game, we are done. If doing that leaves us with no empty stacks, we are also "done", because we have lost the game! So, let's assume that when we begin, there is at least one empty stack.

We will now aim to prove that the following condition is necessary and sufficient for a game to be winnable. Let us construct a graph in which vertices are the suits for which the ace begins the game at the bottom of some stack. We say that a vertex (suit) $s$ is a *source* if the ace is the only card in this suit, and that $s$ is a *target* if there is another ace (of a different suit) in the stack in which the ace of $s$ is at the bottom. We add an edge from vertex $s_1$ to a different vertex $s_2$ if the king of $s_2$ is in the stack that has the ace of $s_1$ at the bottom.

Now, we claim the game is winnable if and only if there exists any path from some source vertex to some target vertex.

To understand this condition, consider a simple case in which there is a single edge from a source suit $s_1$ to a target suit $s_2$; i.e., suit $s_1$ has exactly one card (an ace), which is at the bottom of a stack *A*, and suit $s_2$'s king is in *A*. To ensure that $s_1$ is a source but not a target, assume that there are no other aces in *A*, and to ensure that $s_2$ is a target, assume that its ace is at the bottom of a different stack *B*, and there is a third suit $s_3$ that has an ace higher up in *B*, which we will assume is the ace nearest the bottom except for the bottmmost card in *B*.

The winning strategy in this case is as follows. First, make all legal moves until the only remaining legal move is moving the ace of $s_3$ to an empty pile. Since we won't uncover the ace of $s_1$, there will be fewer than **N** suits visible, so this state is always achievable. When we reach this state, all of the following are true:

- There is an empty stack (since $s_1$ isn't visible, we'd otherwise have two cards visible in one suit, and could remove the one with the lower value).
- Stacks *A* and *B* are the only stacks with more than one card. (Otherwise, we could move a card from from one of the other stacks into the empty space.)
- The other **N**-3 stacks (aside from *A*, *B* and the aforementioned empty stack) each contain an ace of one of the remaining **N**-3 colors. (We couldn't have removed the aces, and they are not in stack *A* or stack *B*).

At this point, we will move the ace of $s_3$ to the empty stack, and then try to remove cards from *B*, until we get down to the ace of $s_2$. If the top card of *B* isn't yet the ace of $s_2$, then it's either in suit $s_2$ (and lower than the king, so we can remove it, because the king is visible), or in some other suit (in which case the ace of that suit is visible, and we can remove it). Therefore, we can remove cards down to the ace of $s_2$, then remove the king of $s_2$, and then again dig down to the ace of $s_1$. We can do this because any card in *A* other than the ace of $s_1$ will be removable, since we now see all the aces other than $s_1$, and there are no cards but the ace in suit $s_1$.

The description above can be extended relatively easily to show how to win the game when a longer path exists. First, we clean up everything but the aces mentioned in the path, and then move the ace from the end of the path into the empty space, and remove all the remaining cards one by one. So, what remains is to prove that if the game is winnable, a path from a source to a target always exists in the graph we constructed.

At the end of a successful game, each of the **N** stacks will contain one of the **N** aces. Whenever we move an ace to the bottom of a stack, it will never again be covered. So, before the last move action, **N**-1 aces will be on the bottom of a stack, and the last move is necesarilly moving an ace to the empty spot. Some of the aces are visible before the last move. Nothing interesting will happen to cards in those suits - we might uncover a card in one of those suits, and then we will be able to immediately remove it, because the ace is visible. The more interesting suits are the ones in which the aces are not visible and are at the bottoms of stacks. Note that once uncovered, an ace cannot be covered again, so these aces had to be at the bottoms of their stacks from the beginning of the game, and the cards on top of them had to be there from the beginning of the game. So, it's enough to prove that we will see a source-target path in the position before the last move. This will mean that the path was there from the beginning of the game.

The cards on top of the other stacks with more than one card have to be in the same set of colors as the covered aces (or else they would have already been removed). We want to prove they are all kings. We will proceed by contradiction: assume that one of them is a lower card (say, a "queen of spades"). Since it is visible and not removed, the king of spades must be somewhere in one of the stacks, and not visible; it cannot have been removed yet, because in this case the ace would have to be visible, and the queen would have been removed as well.

Consider what happens if we move this queen into the empty space. We experience a sequence of removals, which cannot end with removing the queen (that would contradict the assumption that no more moves can be made before the winning one). Thus, it has to end in uncovering the ace of the suit that was not previously visible (causing us to lose the game) - let's call this suit "diamonds".

Now, consider the winning move instead. We also end up with a sequence of removals. After each removal except the last one, we see **N**-1 suits, and so we have exactly one choice what to remove. So, we deterministically remove cards until we, at some point, uncover the king or the ace of spades, whichever comes first, and that causes us to remove the queen of spades... and then execute the exact same deterministic sequence of removals that, in the end, caused us to uncover the ace of diamonds. Note that the other high spade card (whichever among the king and ace that we did not see) is not uncovered in this sequence. If it were, it would have removed the queen of spades in the previous scenario - so, we end up with at least one card that is not visible, which is a contradiction.

So, we have proven that kings are on top of our stacks with (non-visible) aces on the bottom. At this point, following the graph from the ace that was the source will eventually lead us to the target: the stack with two aces.

After establishing all this, the algorithm to check for the desired condition is fairly simple. After constructing our graph, we can start at sources and perform a depth-first search to see if there is a path from any of them to a target. This is considerably faster than running a backtracking algorithm on the set of moves itself, which works for the Small but not for the Large.