Analysis: Irrefutable Outcome

Test Set 1

We can determine the outcome by playing through the game. For each turn: if the current player cannot move, the game is over. If the current player can only move on one side, eliminate the outermost piece from that side. Otherwise, the current player must choose between the leftmost piece and the rightmost piece.

Notice that this choice leads to a subgame where it is the other player's turn. (E.g., the starting game `IOII` where Izabella plays first can lead to either `OII` or `IOI` where it is Olga's turn.) If there is only one piece left in a subgame, the current player wins if they can take the piece and loses if they cannot.

For each turn, eliminate the piece that the current player takes. If the current player can choose between both sides, solve both subgames and choose the option that is more favorable to the current player. Continue until the game has finished.

If N is the length of the input string $\mathbf{B}$, there are at most N moves in the game, and there are at most 2 options per-move, so this solution requires at most $O(2^N)$ time. Moreover, one player having a choice means the pieces at both ends are of their color. Since only one of them is removed, that means the next player cannot have more than one option. This means that at most half of the moves actually have 2 options, meaning our algorithm's running time can be bounded more tightly by $O(2^{N/2})$.

Test Set 2

The solution above can be optimized: use [memoization/dynamic programming](#) to avoid playing a subgame more than once. For each subgame, save its result and if you were to play it again, simply lookup the result instead of doing the recursion. Since there are less than N^2 subgames, this can yield a time complexity as low as $O(N^2)$. Implementations that manipulate strings, use them to represent the states or use inefficient ways to do the lookups can have higher time complexities, but the bounds of this test set are small enough that we can get away with doing it.

Test Set 3

We can alternatively determine the outcome without subgames. For each turn: if the current player cannot move, the game is over. If the current player can only move on one side, eliminate the outermost piece from that side.

If the current player can move on either side, the end state of the game can be calculated as follows: if the `I` and `O` pieces continue to alternate, the last player to take a piece wins (with zero pieces left). Otherwise, the game will end once the players reach a pair of consecutive identical pieces (`II` or `OO`).

Suppose the current player (who can move on either side) is Izabella. From each of the sides, find the nearest pair of consecutive identical pieces (`II` or `OO`). If one of those pairs is `II`, Izabella can win by choosing the outermost piece from that side (as Olga will not be able to move after Izabella takes one of the `I`s in `II`). Note that if both sides have a `II` pair, choosing

the side with the closer pair will maximize Izabella's score — which can be calculated as the number of pieces between that `II` and the outermost piece on the other side, inclusive.

Alternatively, if the nearest pairs on both sides are `OO`, Izabella will ultimately lose the game. She can minimize Olga's score by eliminating the alternating `I` and `O` pieces from both sides until reaching the `OO`(s). Then the score will be the number of pieces between the two occurrences of `OO`s, 1 if there is only one `OO` or 2 if there are overlapping `OO`s, that is, `OOO`, inclusive.

This solution takes linear time.