

Minimum Swaps to Balance Brackets 1 - Solution Outline

Key Ideas

A balanced sequence can be built by cancelling matching pairs from left to right. After removing all pairs, the only useful information is how many unmatched [remain. Each swap can fix two unmatched openings, so the answer is the ceiling of half that count.

Algorithm

1. Scan the string left to right.
2. Maintain a counter for unmatched [.
3. If the character is [, increase the counter.
4. If the character is] and the counter is positive, decrease the counter (a pair is matched).
5. After the scan, let u be the number of unmatched [.
6. Output $(u + 1) // 2$.

Correctness Sketch

Every matched pair does not affect the final balance, so removing them is safe. After removal, the remaining characters must be all [followed by all] in some order. Each swap can correct two misplaced [by bringing in two] to match them, which reduces the unmatched count by 2. Therefore the minimum number of swaps is exactly the smallest integer not less than $u / 2$.

Complexity

- Time: $O(n)$
- Space: $O(1)$