Problem 877: Stone Game

Problem Information

Difficulty: Medium
Acceptance Rate: 72.26%
Paid Only: No
Tags: Array, Math, Dynamic Programming, Game Theory

Problem Description

Alice and Bob play a game with piles of stones. There are an **even** number of piles arranged in a row, and each pile has a **positive** integer number of stones `piles[i]`.

The objective of the game is to end with the most stones. The **total** number of stones across all the piles is **odd** , so there are no ties.

Alice and Bob take turns, with **Alice starting first**. Each turn, a player takes the entire pile of stones either from the **beginning** or from the **end** of the row. This continues until there are no more piles left, at which point the person with the **most stones wins**.

Assuming Alice and Bob play optimally, return `true` _if Alice wins the game, or_`false` _if Bob wins_.

Example 1:

Input: piles = [5,3,4,5] **Output:** true **Explanation:** Alice starts first, and can only take the first 5 or the last 5. Say she takes the first 5, so that the row becomes [3, 4, 5]. If Bob takes 3, then the board is [4, 5], and Alice takes 5 to win with 10 points. If Bob takes the last 5, then the board is [3, 4], and Alice takes 4 to win with 9 points. This demonstrated that taking the first 5 was a winning move for Alice, so we return true.

Example 2:

Input: piles = [3,7,2,3] **Output:** true

Constraints:

* `2 <= piles.length <= 500` * `piles.length` is **even**. * `1 <= piles[i] <= 500` * `sum(piles[i])` is **odd**.

Code Snippets

C++:

```
class Solution {
public:
bool stoneGame(vector<int>& piles) {

}
};
```

Java:

```
class Solution {
public boolean stoneGame(int[] piles) {

}
}
```

Python3:

```
class Solution:
def stoneGame(self, piles: List[int]) -> bool:
```