Problem 1140: Stone Game II

Problem Information

Difficulty: Medium
Acceptance Rate: 72.94%
Paid Only: No
Tags: Array, Math, Dynamic Programming, Prefix Sum, Game Theory

Problem Description

Alice and Bob continue their games with piles of stones. There are a 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.

Alice and Bob take turns, with Alice starting first.

On each player's turn, that player can take **all the stones** in the **first** `X` remaining piles, where `1 <= X <= 2M`. Then, we set `M = max(M, X)`. Initially, M = 1.

The game continues until all the stones have been taken.

Assuming Alice and Bob play optimally, return the maximum number of stones Alice can get.

Example 1:

Input: piles = [2,7,9,4,4]

Output: 10

Explanation:

* If Alice takes one pile at the beginning, Bob takes two piles, then Alice takes 2 piles again. Alice can get `2 + 4 + 4 = 10` stones in total. * If Alice takes two piles at the beginning, then Bob can take all three piles left. In this case, Alice get `2 + 7 = 9` stones in total.

So we return 10 since it's larger.

Example 2:

Input: piles = [1,2,3,4,5,100]

Output: 104

Constraints:

* `1 <= piles.length <= 100` * `1 <= piles[i] <= 104`

Code Snippets

C++:

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

}
};
```

Java:

```
class Solution {
public int stoneGameII(int[] piles) {

}
}
```

Python3:

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