1690. Stone Game VII

Description

Alice and Bob take turns playing a game, with **Alice starting first**.

There are `n` stones arranged in a row. On each player's turn, they can **remove** either the leftmost stone or the rightmost stone from the row and receive points equal to the **sum** of the remaining stones' values in the row. The winner is the one with the higher score when there are no stones left to remove.

Bob found that he will always lose this game (poor Bob, he always loses), so he decided to **minimize the score's difference**. Alice's goal is to **maximize the difference** in the score.

Given an array of integers `stones` where `stones[i]` represents the value of the `i th` stone **from the left**, return *the **difference** in Alice and Bob's score if they both play **optimally**.*

Example 1:

```
Input: stones = [5,3,1,4,2]
Output: 6
Explanation:
- Alice removes 2 and gets 5 + 3 + 1 + 4 = 13 points. Alice = 13, Bob = 0, stones = [5,3,1,4].
- Bob removes 5 and gets 3 + 1 + 4 = 8 points. Alice = 13, Bob = 8, stones = [3,1,4].
- Alice removes 3 and gets 1 + 4 = 5 points. Alice = 18, Bob = 8, stones = [1,4].
- Bob removes 1 and gets 4 points. Alice = 18, Bob = 12, stones = [4].
- Alice removes 4 and gets 0 points. Alice = 18, Bob = 12, stones = [].
The score difference is 18 - 12 = 6.
```

Example 2:

```
Input: stones = [7,90,5,1,100,10,10,2]
Output: 122
```

Constraints:

- `n == stones.length`
- `2 <= n <= 1000`
- `1 <= stones[i] <= 1000`