Problem 1040: Moving Stones Until Consecutive II

Problem Information

Difficulty: Medium
Acceptance Rate: 58.30%
Paid Only: No
Tags: Array, Math, Sliding Window, Sorting

Problem Description

There are some stones in different positions on the X-axis. You are given an integer array `stones`, the positions of the stones.

Call a stone an **endpoint stone** if it has the smallest or largest position. In one move, you pick up an **endpoint stone** and move it to an unoccupied position so that it is no longer an **endpoint stone**.

* In particular, if the stones are at say, `stones = [1,2,5]`, you cannot move the endpoint stone at position `5`, since moving it to any position (such as `0`, or `3`) will still keep that stone as an endpoint stone.

The game ends when you cannot make any more moves (i.e., the stones are in three consecutive positions).

Return _an integer array_`answer` _of length_`2` _where_ :

* `answer[0]` _is the minimum number of moves you can play, and_ * `answer[1]` _is the maximum number of moves you can play_.

Example 1:

Input: stones = [7,4,9] **Output:** [1,2] **Explanation:** We can move 4 -> 8 for one move to finish the game. Or, we can move 9 -> 5, 4 -> 6 for two moves to finish the game.

Example 2:

Input: stones = [6,5,4,3,10] **Output:** [2,3] **Explanation:** We can move 3 -> 8 then 10 -> 7 to finish the game. Or, we can move 3 -> 7, 4 -> 8, 5 -> 9 to finish the game. Notice we cannot move 10 -> 2 to finish the game, because that would be an illegal move.

Constraints:

* `3 <= stones.length <= 104` * `1 <= stones[i] <= 109` * All the values of `stones` are **unique**.

Code Snippets

C++:

```
class Solution {
public:
vector<int> numMovesStonesII(vector<int>& stones) {

}
};
```

Java:

```
class Solution {
public int[] numMovesStonesII(int[] stones) {

}
}
```

Python3:

```
class Solution:
def numMovesStonesII(self, stones: List[int]) -> List[int]:
```