Strictly Increasing Array

Given an array **nums** of **n** positive integers. Find the minimum number of operations required to modify the array such that array elements are in **strictly increasing** order (nums[i] < nums[i+1]).
Changing a number to **greater** or **lesser** than original number is counted as one operation.

Note: Array elements can become negative after applying operation.

Example 1:

```
Input:
n = 6
nums = [1, 2, 3, 6, 5, 4]
Output:
2
Explanation:
By decreasing 6 by 2 and increasing 4 by 2, nums will
be like [1, 2, 3, 4, 5, 6] which is stricly increasing.
```

Example 2:

```
Input:
n = 4
nums = [1, 1, 1, 1]
Output:
3
Explanation:
One such array after operation can be [-2, -1, 0, 1].
We require atleast 3 operations for this example.
```

Your Task:
You don't need to read or print anything. Your task is to complete the function **min_opeartions()** which takes the array **nums** as input parameter

and returns the minimum number of opeartion needed to make the array strictly increasing.

Expected Time Complexity: O(n^2)
Expected Space Complexity: O(n)

Constraints:
$1 <= n <= 10^3$
$1 <= nums[i] <= 10^9$