Longest Continuous Subarray With Absolute Diff Less Than or Equal to Limit

Given an array of integers nums and an integer limit, return the size of the longest non-empty subarray such that the absolute difference between any two elements of this subarray is less than or equal to limit.

Example 1:

Input: nums = [8,2,4,7], limit = 4

Output: 2

Program:

```
from collections import deque
def longest_subarray(nums, limit):
    max_deque = deque()
    min_deque = deque()
    left = 0
    max_len = 0
    for right in range(len(nums)):
        while max_deque and nums[max_deque[-1]]
<= nums[right]:
            max_deque.pop()
        while min_deque and nums[min_deque[-
1]] >= nums[right]:
            min_deque.pop()
        max_deque.append(right)
        min_deque.append(right)
```

```
        while nums[max_deque[0]] -
nums[min_deque[0]] > limit:
            left += 1
            if max_deque[0] < left:
                max_deque.popleft()
            if min_deque[0] < left:
                min_deque.popleft()
        max_len = max(max_len, right - left + 1)
    return max_len
nums = [8, 2, 4, 7]
limit = 4
print(longest_subarray(nums, limit))
```

Output:


Time complexity:

O(n)