LONGEST CONTINUOUS SUBARRAY

```
class Solution {
    public int longestSubarray(int[] nums, int limit) {
        Deque<Integer> maxDeque = new LinkedList<>();
        Deque<Integer> minDeque = new LinkedList<>();
        int left = 0;
        int ans = 0;
        for (int right = 0; right < nums.length; right++) {
            while (!maxDeque.isEmpty() && nums[right] > maxDeque.peekLast()) {
                maxDeque.pollLast();
            }
            maxDeque.addLast(nums[right]);
            while (!minDeque.isEmpty() && nums[right] < minDeque.peekLast()) {
                minDeque.pollLast();
            }
            minDeque.addLast(nums[right]);
            while (maxDeque.peekFirst() - minDeque.peekFirst() > limit) {
                if (nums[left] == maxDeque.peekFirst()) {
                    maxDeque.pollFirst();
                }
                if (nums[left] == minDeque.peekFirst()) {
                    minDeque.pollFirst();
                }
                left++;
            }
            ans = Math.max(ans, right - left + 1);
        }
        return ans;
    }
}
```