2. 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.

Code:

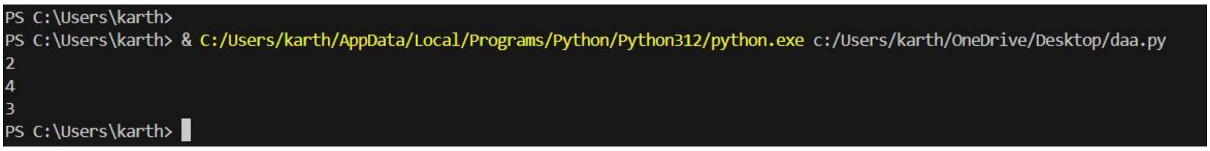
```
from collections import deque

def longestSubarray(nums,limit):

   max_deque=deque()

   min_deque=deque()

   left=0

   max_length=0

   for right in range(len(nums)):

      while max_deque and nums[max_deque[-1]]<=nums[right]:

         max_deque.pop()

      max_deque.append(right)

      while min_deque and nums[min_deque[-1]]>=nums[right]:

         min_deque.pop()

      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_length=max(max_length,right-left + 1)

   return max_length

nums=[8,2,4,7]

limit=4

print(longestSubarray(nums,limit))

nums=[10, 1, 2, 4, 7, 2]

limit=5

print(longestSubarray(nums,limit))
```

nums=[4, 2, 2, 2, 4, 4, 2, 2]

limit=0

print(longestSubarray(nums,limit)

output:



```
PS C:\Users\karth>
PS C:\Users\karth> & C:/Users/karth/AppData/Local/Programs/Python/Python312/python.exe c:/Users/karth/OneDrive/Desktop/daa.py
2
4
3
PS C:\Users\karth>
```

Time complexity:

F(n)=o(n)