

52) Max Chunks To Make Sorted You are given an integer array `arr` of length `n` that represents a permutation of the integers in the range `[0, n - 1]`. We split `arr` into some number of chunks (i.e., partitions), and individually sort each chunk. After concatenating them, the result should equal the sorted array. Return the largest number of chunks we can make to sort the array.

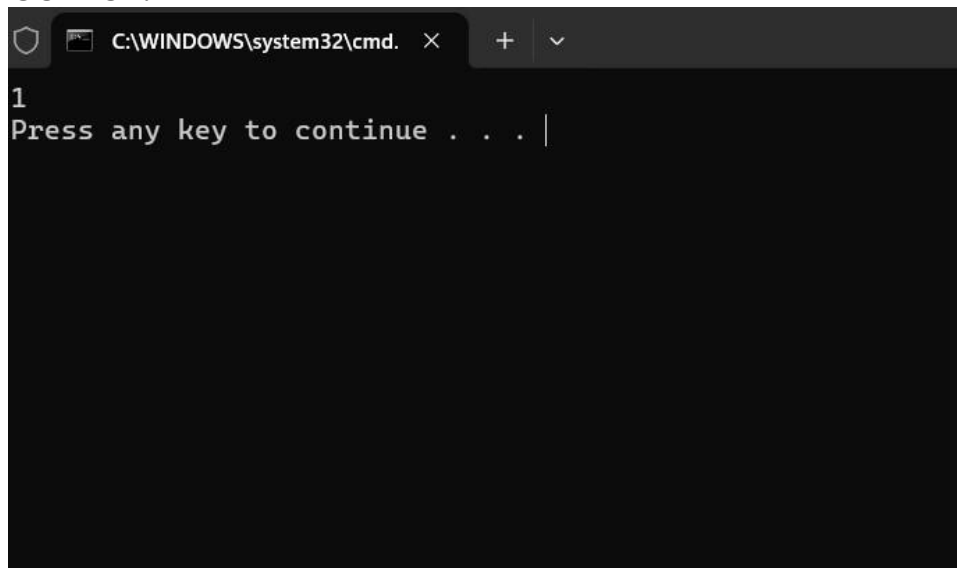
CODE: def

```
max_chunks_to_make_sorted(arr):
    max_so_far = -1    chunks = 0
    for i in range(len(arr)):
        max_so_far = max(max_so_far, arr[i])    if
    max_so_far == i:        chunks += 1

    return chunks
```

```
arr = [4, 3, 2, 1, 0]
print(max_chunks_to_make_sorted(arr))
```

OUTPUT:

A screenshot of a Windows command prompt window. The title bar shows the path 'C:\WINDOWS\system32\cmd.' and standard window controls. The command prompt displays the number '1' on the first line, followed by the text 'Press any key to continue . . . |' on the second line, with a vertical cursor at the end.

TIME COMPLEXITY : $O(n)$