Length of the largest subarray with contiguous elements

Given an array of integers, find length of the longest subarray which contains numbers that can be arranged in a continuous sequence.
In the previous post, we have discussed a solution that assumes that elements in given array are distinct. Here we discuss a solution that works even if the input array has duplicates.
Examples:

```
Input:  arr[] = {10, 12, 11};
Output: Length of the longest contiguous subarray is 3

Input:  arr[] = {10, 12, 12, 10, 10, 11, 10};
Output: Length of the longest contiguous subarray is 2
```

```python
def largestContiguousArray(arr):
    ans = 0
    myset = set()
    n = len(arr)
    for i in range(n - 1):
        maxEl = arr[i]
        minEl = arr[i]
        myset.add(arr[i])
        for j in range(i+1, n):
            if arr[j] in myset:
                break
            else:
                myset.add(arr[j])
                maxEl = max(maxEl, arr[j])
                minEl = min(minEl, arr[j])
                if maxEl - minEl == j - i:
                    ans = max(ans, j - i + 1)
    return ans



arr = [10, 12, 12, 10, 10, 11, 10]
print(largestContiguousArray(arr))
```