Arrays Assignment_29_Oct

November 3, 2023

[2]:

```
#1. Given an array, check if it contains any duplicates or not.
#arr = [1, 2, 4, 2, 5, 9]
#Output = True1.

def contains_duplicates(arr):
    seen = set()
    for num in arr:
        if num in seen:
            return True
        seen.add(num)
    return False

arr = [1, 2, 4, 2, 5, 9]
output = contains_duplicates(arr)
print(output)
```

```
True
```

[3]:

```
#2. Given an array and an integer k, rotate the array to the right by k steps.
#arr = [1, 2, 3, 4, 5, 6, 7] k = 3
#Output = [5, 6, 7, 1, 2, 3, 4]

def rotate_array(arr, k):
    if not arr:
        return arr

    n = len(arr)
    k = k % n  # In case k is larger than the array size, take the modulo to␣
 ↪find the effective rotation.

    if k == 0:
        return arr

    # Reverse the entire array.
    reverse(arr, 0, n - 1)
    # Reverse the first k elements.
    reverse(arr, 0, k - 1)
    # Reverse the remaining elements.
    reverse(arr, k, n - 1)

def reverse(arr, start, end):
    while start < end:
        arr[start], arr[end] = arr[end], arr[start]
        start += 1
        end -= 1

arr = [1, 2, 3, 4, 5, 6, 7]
k = 3
rotate_array(arr, k)
print(arr)
```

```
[5, 6, 7, 1, 2, 3, 4]
```

[4]:

```
#3. Reverse the given array in-place, means without using any extra data␣
 ↪structure.
#arr = [2, 4, 5, 7, 9, 12]
#Output = [12, 9, 7, 5, 4, 2]

def reverse_array_in_place(arr):
    start = 0
    end = len(arr) - 1

    while start < end:
        arr[start], arr[end] = arr[end], arr[start]  # Swap elements at the␣
 ↪start and end positions.
        start += 1
        end -= 1

arr = [2, 4, 5, 7, 9, 12]
reverse_array_in_place(arr)
print(arr)
```

```
[12, 9, 7, 5, 4, 2]
```

[5]:

```
#4. Given an array of integers, find the maximum element in an array
#arr = [10, 5, 20, 8, 15]
#Output = 20

def find_maximum(arr):
    if not arr:
        return None

    max_element = arr[0]

    for num in arr:
        if num > max_element:
            max_element = num

    return max_element

arr = [10, 5, 20, 8, 15]
output = find_maximum(arr)
print(output)
```

```
20
```

[6]:

```
#5. Given a sorted array, remove the duplicate element without using any extra␣
  ↪data structure.
#arr = [1, 1, 2, 2, 2, 3, 3, 4, 4, 4, 5, 5]
#Output = [1, 2, 3, 4, 5]



def remove_duplicates(arr):
    if not arr:
        return arr

    # Initialize the index for the next unique element.
    unique_index = 0

    for i in range(1, len(arr)):
        if arr[i] != arr[unique_index]:
            unique_index += 1
            arr[unique_index] = arr[i]

    # Slice the array to retain only the unique elements.
    arr[:] = arr[:unique_index + 1]

arr = [1, 1, 2, 2, 2, 3, 3, 4, 4, 4, 5, 5]
remove_duplicates(arr)
print(arr)
```

```
[1, 2, 3, 4, 5]
```

[ ]:

```

```