

## Week4Q1

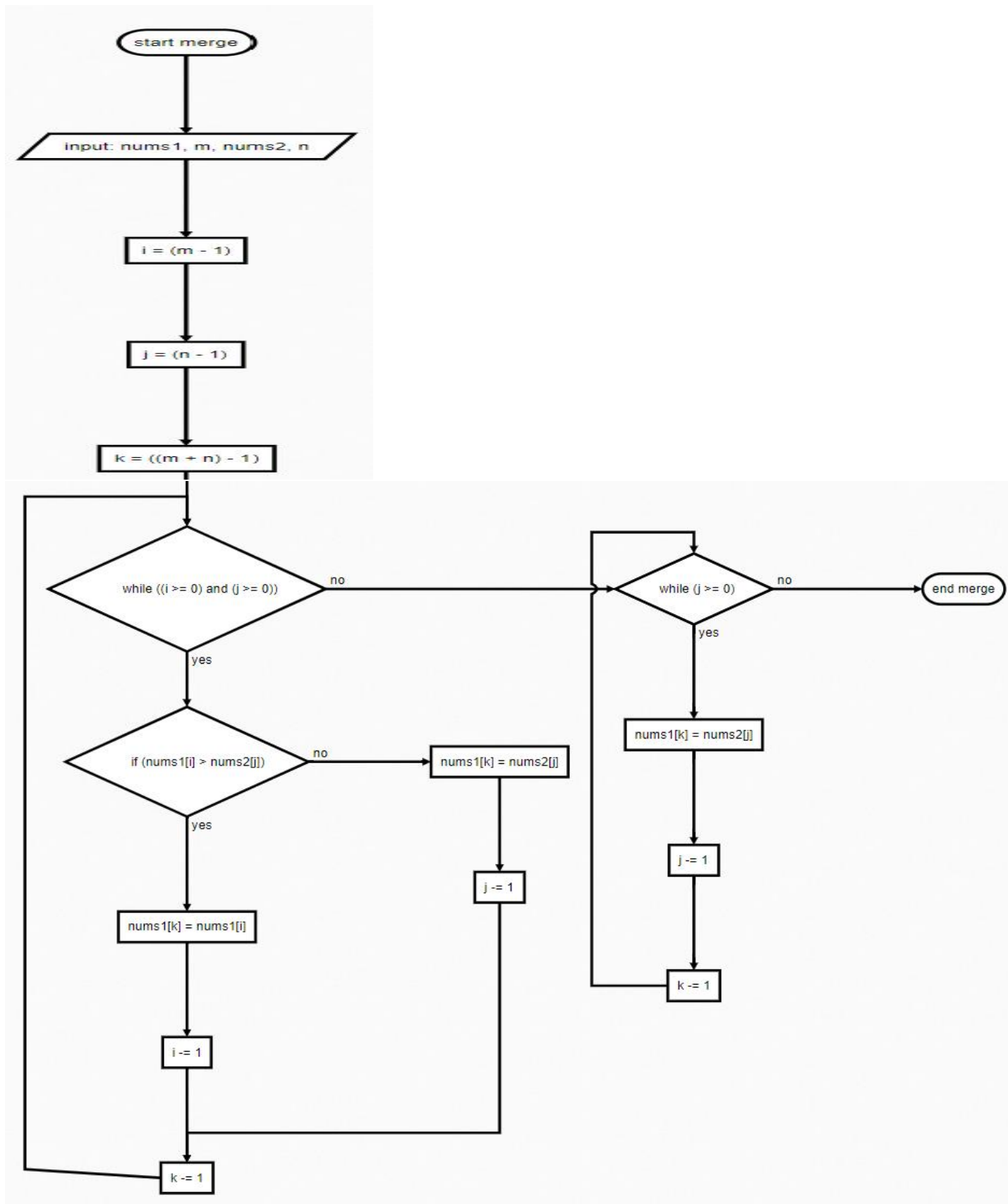
Python code

```
def merge(nums1, m, nums2, n):
    # Initialize pointers for nums1, nums2, and the merged array
    i = m - 1 # Pointer for nums1
    j = n - 1 # Pointer for nums2
    k = m + n - 1 # Pointer for the merged array

    # Iterate from the end of the arrays
    while i >= 0 and j >= 0:
        if nums1[i] > nums2[j]:
            # If element in nums1 is larger, place it in the merged array
            nums1[k] = nums1[i]
            i -= 1
        else:
            # If element in nums2 is larger, place it in the merged array
            nums1[k] = nums2[j]
            j -= 1
        k -= 1

    # If there are any remaining elements in nums2, append them to nums1
    while j >= 0:
        nums1[k] = nums2[j]
        j -= 1
        k -= 1

nums1 = [1, 2, 0, 0]
m = 2
nums2 = [2, 3]
n = 2
merge(nums1, m, nums2, n)
print(nums1)
```



## Tractable

[illegible]

## Testcases

```
def merge(nums1, m, nums2, n):
    # Initialize pointers for nums1, nums2, and the merged array
    i = m - 1 # Pointer for nums1
    j = n - 1 # Pointer for nums2
    k = m + n - 1 # Pointer for the merged array

    # Iterate from the end of the arrays
    while i >= 0 and j >= 0:
        if nums1[i] > nums2[j]:
            # If element in nums1 is larger, place it in the merged array
            nums1[k] = nums1[i]
            i -= 1
        else:
            # If element in nums2 is larger, place it in the merged array
            nums1[k] = nums2[j]
            j -= 1
        k -= 1

    # If there are any remaining elements in nums2, append them to nums1
    while j >= 0:
        nums1[k] = nums2[j]
        j -= 1
        k -= 1

nums1 = [1, 2, 0, 0]
m = 2
nums2 = [2, 3]
n = 2
merge(nums1, m, nums2, n)
print(nums1)

nums1a=[1,2,3,0,0,0]
m1=3
nums2a=[2,5,6]
n1=3
merge(nums1a,m1,nums2a,n1)
print(nums1a)

nums1b=[1]
m2=1
```

```

nums2b=[]
n2=0
merge(nums1b,m2,nums2b,n2)
print(nums1b)

```

```

nums1c=[0]
m3=0
nums2c=[1]
n3=1
merge(nums1c,m3,nums2c,n3)
print(nums1c)

```

```

1 def merge(nums1, m, nums2, n):
2     # Initialize pointers for nums1, nums2, and the merged array
3     i = m - 1 # Pointer for nums1
4     j = n - 1 # Pointer for nums2
5     k = m + n - 1 # Pointer for the merged array
6
7     # Iterate from the end of the arrays
8     while i >= 0 and j >= 0:
9         if nums1[i] > nums2[j]:
10             # If element in nums1 is larger, place it in the merged array
11             nums1[k] = nums1[i]
12             i -= 1
13         else:
14             # If element in nums2 is larger, place it in the merged array
15             nums1[k] = nums2[j]
16             j -= 1
17             k -= 1
18
19     # If there are any remaining elements in nums2, append them to nums1
20     while j >= 0:
21         nums1[k] = nums2[j]
22         j -= 1
23         k -= 1
24
25
26 nums1 = [1, 2, 0, 0]

```

[Running] python -u "c:\Users\cheth\OneDrive\Desktop\SFBU\SEM3\ALGORITHMS\week4q1.py"

[1, 2, 2, 3]  
[1, 2, 2, 3, 5, 6]  
[1]  
[1]

[Done] exited with code=0 in 0.186 seconds

Activate Windows  
Go to Settings to activate Windows.

Ln 15, Col 32 Spaces: 4 UTF-8 CRLF Python 3.9.13 64-bit (microsoft store)

Visual Studio Code interface showing a Python file named `week4q1.py` with a merge function implementation. The Explorer sidebar lists various files, including `ALGORITHMS` and `week4q1.py`. The Output window shows the execution of the script, displaying the output of the `merge` function.

```
20 while j >= 0:
21     nums1[k] = nums2[j]
22     j -= 1
23     k -= 1
24
25
26 nums1 = [1, 2, 0, 0]
27 m = 2
28 nums2 = [2, 3]
29 n = 2
30 merge(nums1, m, nums2, n)
31 print(nums1)
32
33 nums1a=[1,2,3,0,0,0]
34 m1=3
35 nums2a=[2,5,6]
36 n1=3
37 merge(nums1a,m1,nums2a,n1)
38 print(nums1a)
39
40
41 nums1b=[1]
42 m2=1
43 nums2b=[]
44 n2=0
```

Output:

```
[Running] python -u "c:\Users\cheth\OneDrive\Desktop\SFBUS\SEM3\ALGORITHMS\week4q1.py"
[1, 2, 2, 3]
[1, 2, 2, 3, 5, 6]
[1]
[1]
[Done] exited with code=0 in 0.186 seconds
```

Windows taskbar shows the date and time as 4:37 PM on 6/10/2023.