```python
def merge_sorted_array(nums1, m, nums2, n):
    ptr1 = m - 1
    ptr2 = n - 1
    merged_ptr = m + n - 1

    while ptr1 >= 0 and ptr2 >= 0:
        if nums1[ptr1] > nums2[ptr2]:
            nums1[merged_ptr] = nums1[ptr1]
            ptr1 -= 1
        else:
            nums1[merged_ptr] = nums2[ptr2]
            ptr2 -= 1
        merged_ptr -= 1

    while ptr2 >= 0:
        nums1[merged_ptr] = nums2[ptr2]
        ptr2 -= 1
        merged_ptr -= 1

    return nums1

# Test case
nums1 = [1, 2, 0, 0]
m = 2
nums2 = [2, 3]
n = 2

result = merge_sorted_array(nums1, m, nums2, n)
print(result)
```
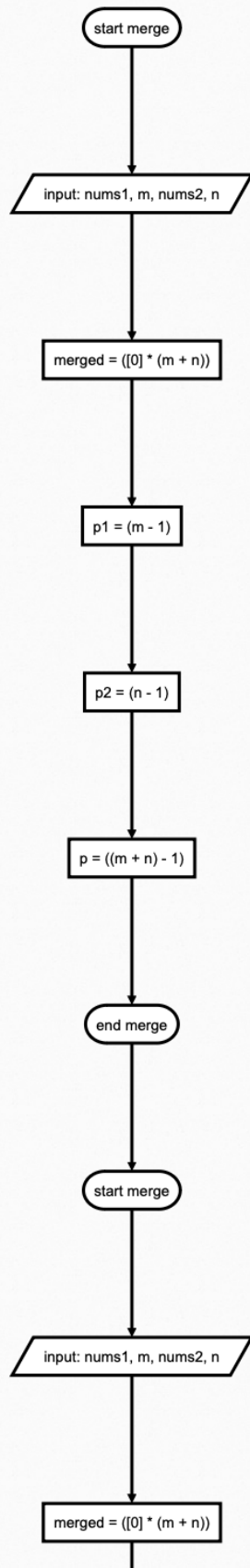
```
start merge
```

input: nums1, m, nums2, n

merged = ([0] * (m + n))

p1 = (m - 1)

p2 = (n - 1)

p = ((m + n) - 1)

```
end merge
```

```
start merge
```

input: nums1, m, nums2, n

merged = ([0] * (m + n))

--

Test cases:

```
def merge(nums1, m, nums2, n):

    ptr1 = m − 1

    ptr2 = n − 1

    merged_ptr = m + n − 1


    while ptr1 >= 0 and ptr2 >= 0:
```

```python
        if nums1[ptr1] > nums2[ptr2]:

            nums1[merged_ptr] = nums1[ptr1]

            ptr1 -= 1

        else:

            nums1[merged_ptr] = nums2[ptr2]

            ptr2 -= 1

        merged_ptr -= 1


    while ptr2 >= 0:

        nums1[merged_ptr] = nums2[ptr2]

        ptr2 -= 1

        merged_ptr -= 1


    return nums1



# Test case 1

nums1a = [1, 2, 3, 0, 0, 0]

m1 = 3

nums2a = [2, 5, 6]

n1 = 3


merge(nums1a, m1, nums2a, n1)

print(nums1a)  # Output: [1, 2, 2, 3, 5, 6]
```
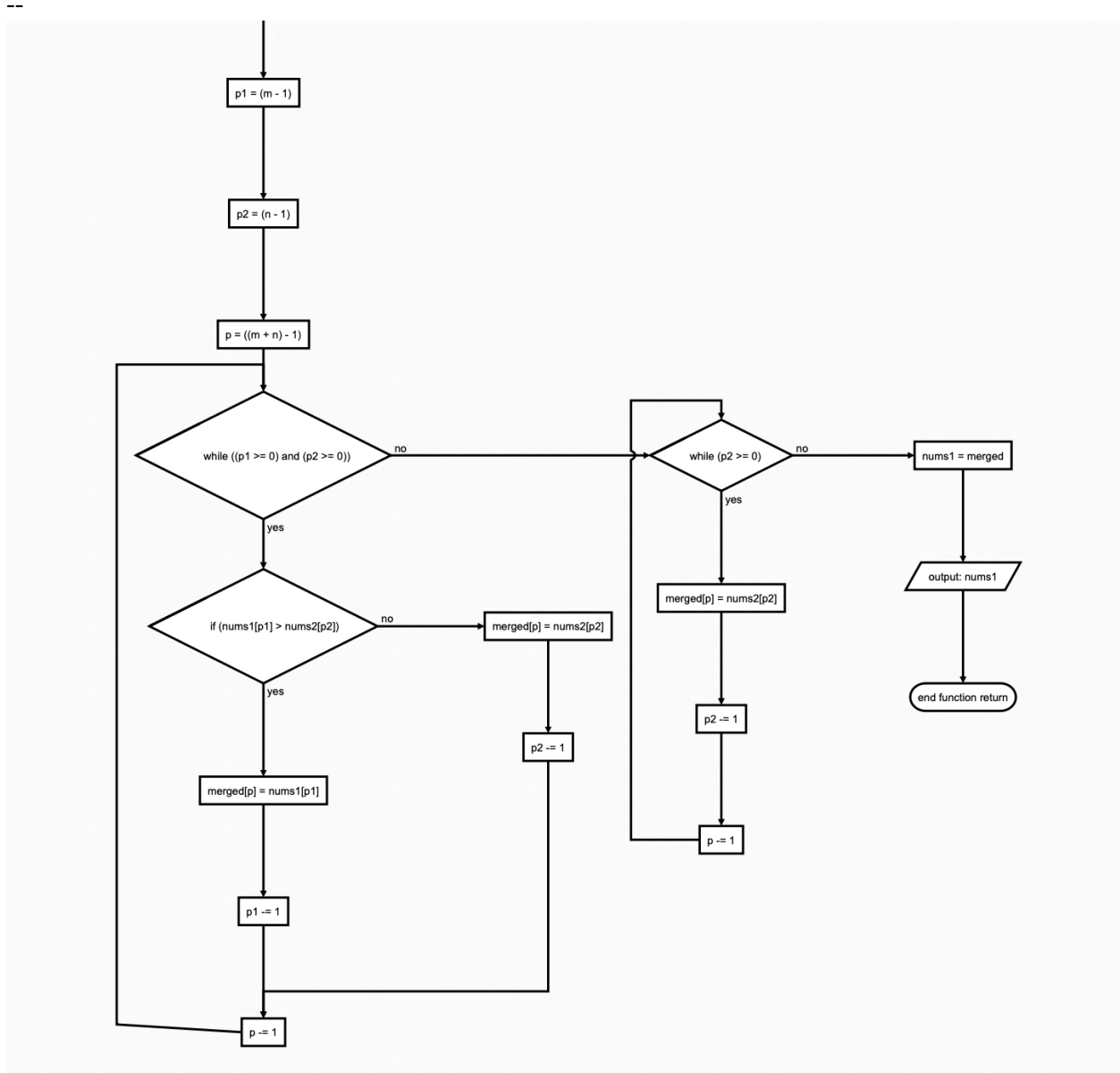
```
# Test case 2

nums1b = [1]

m2 = 1

nums2b = []

n2 = 0


merge(nums1b, m2, nums2b, n2)

print(nums1b)  # Output: [1]
```

```
# Test case 3

nums1c = [0]

m3 = 0

nums2c = [1]

n3 = 1


merge(nums1c, m3, nums2c, n3)

print(nums1c)  # Output: [1]
```

Line	Nums1	M	Nums2	N	Ptr1	Ptr2	Merged_ptr	Ptr1 >=0	Ptr2 >=0	Ptr1>=0 &ptr2 >=0	Nums[ptr1]>nums2[ptr2	return
1	[1,2,0,0]	2	[2,3]	2								

2					2−1=1						
3					2−1=1						
4						2+2−1=3					
6									true		
7										2>3=>false	
11	[1,2,0,3]										
12					1−1=0						
13						3−1=2					
6									true		
7										2>2=>false	
11	[1,2,2,3]										
12					0−1=−1						
13						2−1=1					
15								false			
20											[1,2,2,3]

```python
36      nums2b = []
37      n2 = 0
38
39      merge(nums1b, m2, nums2b, n2)
40      print(nums1b)  # Output: [1]
41
42
43      # Test case 3
44      nums1c = [0]
45      m3 = 0
46      nums2c = [1]
47      n3 = 1
48
49      merge(nums1c, m3, nums2c, n3)
50      print(nums1c)  # Output: [1]
```

PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL

```
/usr/local/bin/python3 /Users/rakesh_kasha/Desktop/week5_Q1.py
(base) rakesh_kasha@Rakesh-kashas-MacBook-Pro ~ % /usr/local/bin/python3 /Users/rakesh_kasha/Desktop/week5_Q1.py
[0, 0, 2, 3]
[0, 0, 2, 3, 5, 6]
[0]
[1]
(base) rakesh_kasha@Rakesh-kashas-MacBook-Pro ~ % /usr/local/bin/python3 /Users/rakesh_kasha/Desktop/week5_q2.py
2
1
4
(base) rakesh_kasha@Rakesh-kashas-MacBook-Pro ~ % /usr/local/bin/python3 /Users/rakesh_kasha/Desktop/week5_Q1.py
[1, 2, 2, 3, 5, 6]
[1]
[1]
(base) rakesh_kasha@Rakesh-kashas-MacBook-Pro ~ %
```