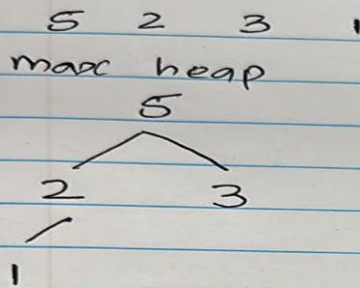


Manual solution

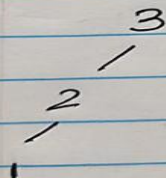


after max heap arr = [5, 2, 3, 1]

solution

replace 5 with 3

5 goes to last position in array



arr = [, , , 5]

replace 3 with 2 3 goes to next last position



arr = [, 3, 5]

~~replace~~ 2 is in root position.

arr = [, 2, 3, 5] \Rightarrow [1, 2, 3, 5]

Chatgpt code

```
def heapify(nums, n, i):
    largest = i # Initialize the largest element as the root
    left = 2 * i + 1 # Left child
    right = 2 * i + 2 # Right child

    # Check if the left child exists and is greater than the root
    if left < n and nums[i] < nums[left]:
        largest = left

    # Check if the right child exists and is greater than the largest element so far
    if right < n and nums[largest] < nums[right]:
        largest = right

    # If the largest element is not the root, swap them and heapify the affected subtree
    if largest != i:
        nums[i], nums[largest] = nums[largest], nums[i]
        heapify(nums, n, largest)

def heapSort(nums):
    n = len(nums)

    # Build a max heap
    for i in range(n // 2 - 1, -1, -1):
        heapify(nums, n, i)

    # Extract elements one by one from the heap
    for i in range(n - 1, 0, -1):
        nums[i], nums[0] = nums[0], nums[i] # Swap the root (max element) with the last element
        heapify(nums, i, 0) # Heapify the reduced heap

    return nums

nums = [5, 2, 3, 1]
output = heapSort(nums)
print(output)
```

test case

The screenshot shows a Visual Studio Code window with the file explorer on the left, the editor in the center, and the terminal at the bottom. The file explorer shows a project named 'ALGORITHMS' with various files including test cases and flowcharts. The editor displays a Python script for heap sort. The terminal shows the command to run the script and its output, which matches the test cases provided in the code.

```
File Edit Selection View Go Run Terminal Help
week9hw2.py - ALGORITHMS - Visual Studio Code

EXPLORER
ALGORITHMS
week4q1.py
week4q1flowchartpt...
week4q1flowchart.html
week4q1flowchartp1...
week4q1flowchartp2...
week4q1flowchartpar...
week4q1testcase1.png
week4q1testcase2.png
Week4q2.docx
week4q2.png
week4q2.py
week4q2testcase1.png
week4q2testcase2.png
week4q3.py
week4q3flowchart.png
week4q3testcase.png
Week4q21.docx
week6q1.py
week6q1p1.png
week6q1p2.png
week6q2.py
week6q2p1.png
week6q2p2.png
week6q3.py
WEEK6Q3P1.png
WEEK6Q3P2.png
week9hw1.png
week9hw1.py
week9hw2.png
week9hw2.py
OUTLINE
TIMELINE

week9hw2.py > heapSort
1 def heapify(nums, n, i):
2     largest = i # Initialize the largest element as the root
3     left = 2 * i + 1 # Left child
4     right = 2 * i + 2 # Right child
5
6     # Check if the left child exists and is greater than the root
7     if left < n and nums[left] < nums[largest]:
8         largest = left
9
10    # Check if the right child exists and is greater than the largest element so far
11    if right < n and nums[right] < nums[largest]:
12        largest = right
13
14    # If the largest element is not the root, swap them and heapify the affected subtree
15    if largest != i:
16        nums[i], nums[largest] = nums[largest], nums[i]
17        heapify(nums, n, largest)
18 def heapSort(nums):
19     n = len(nums)
20     for i in range(n // 2 - 1, -1, -1):
21         heapify(nums, n, i)
22     for i in range(n - 1, 0, -1):
23         nums[i], nums[0] = nums[0], nums[i] # Swap the root (max element) with the last element
24         heapify(nums, i, 0) # Heapify the reduced heap
25     return nums
26 nums = [5, 2, 3, 1]
27 output = heapSort(nums)
28 print('result of first test case',output)
29 nums = [5,1,1,2,0,0]
30 output = heapSort(nums)
31 print('result of second test case',output)

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Code
[Running] python -u "c:\Users\cheth\OneDrive\Desktop\SFBU\SEM3\ALGORITHMS\week9hw2.py"
result of first test case [1, 2, 3, 5]
result of second test case [0, 0, 1, 1, 2, 5]

Ln 24, Col 56 Spaces: 4 UTF-8 CRLF Python 3.9.13 64-bit (microsoft store)
5:29 PM 7/13/2023
```