United International University (UIU)

Final Exam Preparation for Fall 2024

Course: Data Structures & Algorithms – 1 (DSA 1)

Topic: Heap, Heap Sort, Heapify

Learn With Mahfuz

Heap, Heap Sort, Heapify

Spring - 24

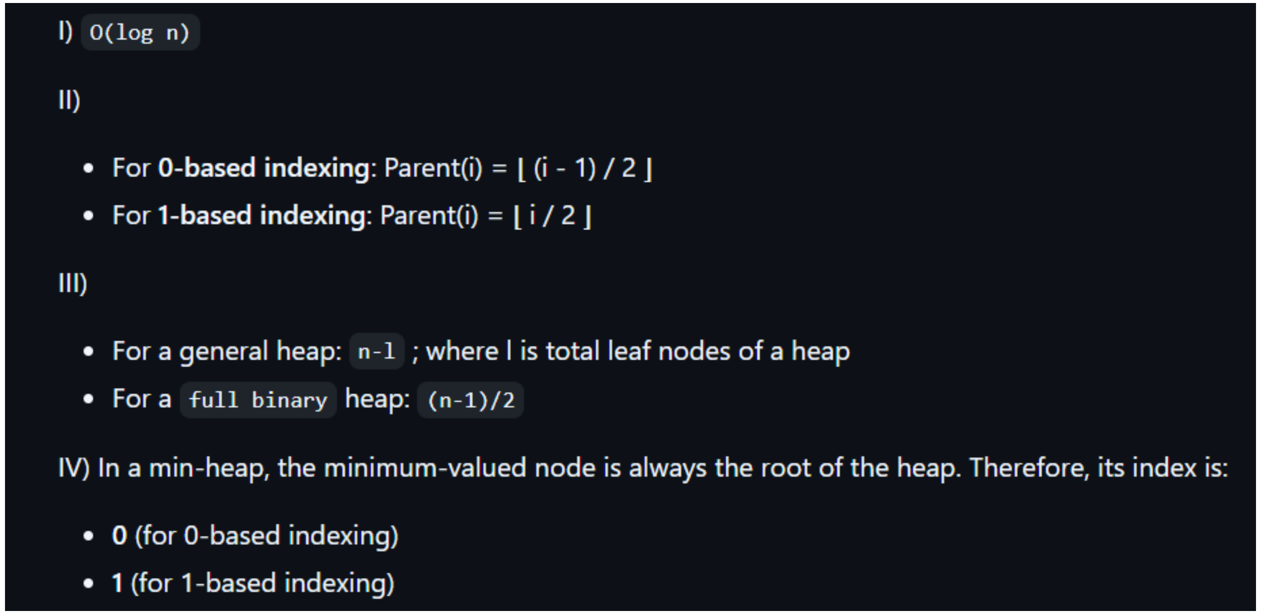
3. (a) Considering a binary heap containing n nodes, answer the following in terms of n: [0.5*4=2]

I. What is the time complexity of heapify()?

II. Write down the index of the parent of node ***i*** in a heap.

III. How many internal nodes does a heap have?

IV. Write down the index of the minimum valued node in a minheap.



I) `O(log n)`

II)

- For **0-based indexing**: Parent(i) = ⌊ (i - 1) / 2 ⌋
- For **1-based indexing**: Parent(i) = ⌊ i / 2 ⌋

III)

- For a general heap: `n-l` ; where l is total leaf nodes of a heap
- For a `full binary` heap: `(n-1)/2`

IV) In a min-heap, the minimum-valued node is always the root of the heap. Therefore, its index is:

- **0** (for 0-based indexing)
- **1** (for 1-based indexing)

Learn With Mahfuz

Let's Code Together

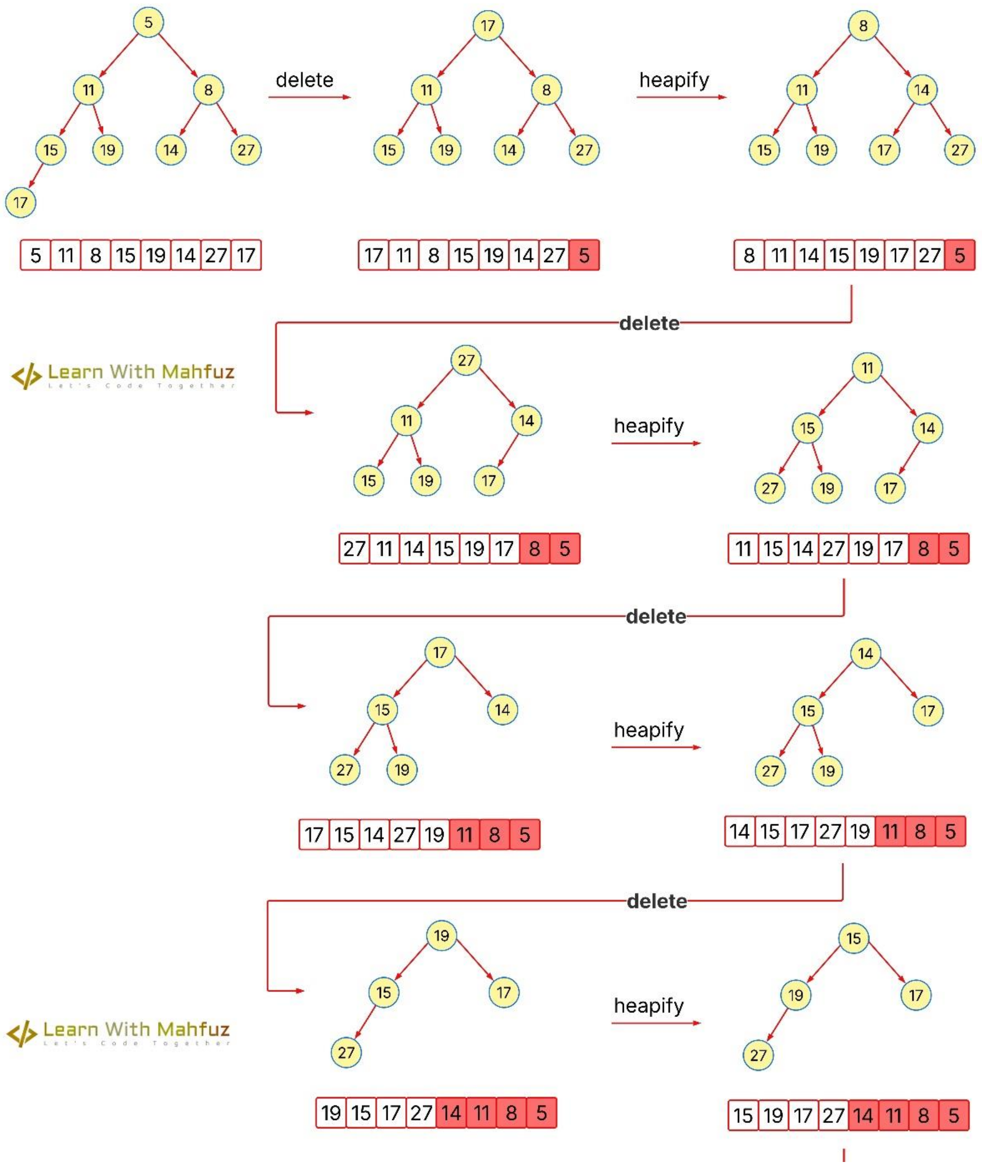
Heap, Heap Sort, Heapify

Spring - 24

(b) Show the steps of heapsort in descending order for the following numbers. The given heap is a Minheap. Just show the steps of sorting. Draw separate trees for each step and write down the final sorted sequence. [3]

5, 11, 8, 15, 19, 14, 27, 17

Spring - 24



5 11 8 15 19 14 27 17

17 11 8 15 19 14 27 5

8 11 14 15 19 17 27 5

27 11 14 15 19 17 8 5

11 15 14 27 19 17 8 5

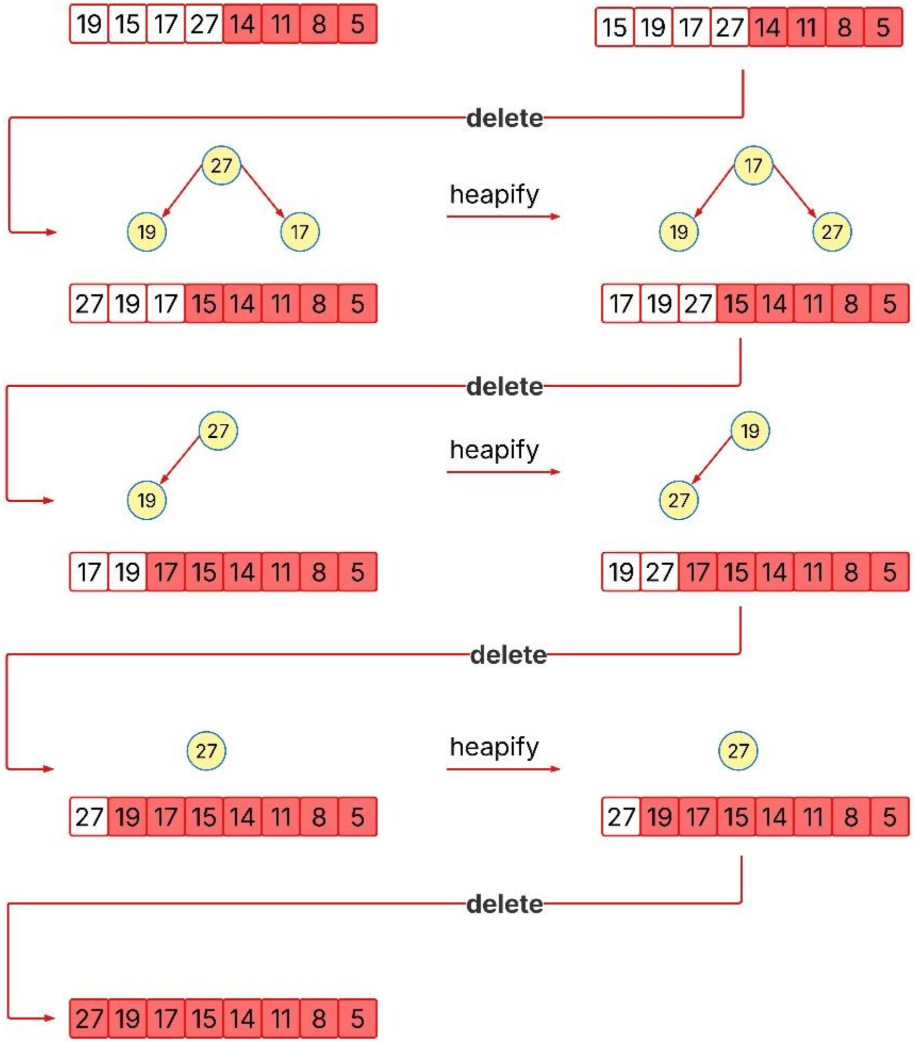
17 15 14 27 19 11 8 5

14 15 17 27 19 11 8 5

19 15 17 27 14 11 8 5

15 19 17 27 14 11 8 5

Spring - 24



19 15 17 27 14 11 8 5

15 19 17 27 14 11 8 5

delete

27 19 17 15 14 11 8 5

heapify

17 19 27 15 14 11 8 5

delete

17 19 17 15 14 11 8 5

heapify

19 27 17 15 14 11 8 5

delete

27 19 17 15 14 11 8 5

heapify

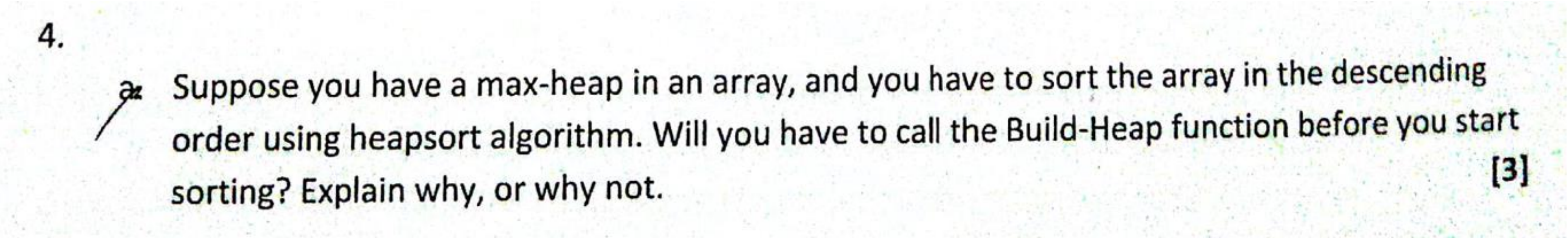
27 19 17 15 14 11 8 5

delete

27 19 17 15 14 11 8 5

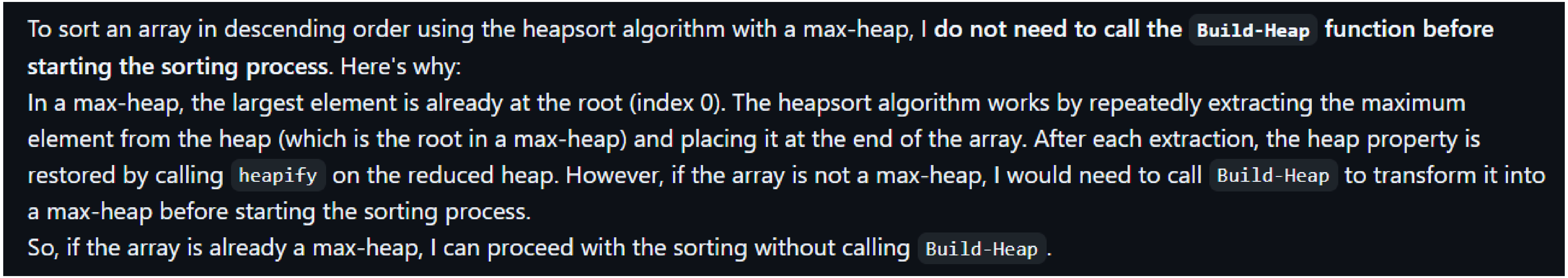
Heap, Heap Sort, Heapify

Summer - 24



4.

~~a~~ Suppose you have a max-heap in an array, and you have to sort the array in the descending order using heapsort algorithm. Will you have to call the Build-Heap function before you start sorting? Explain why, or why not. [3]



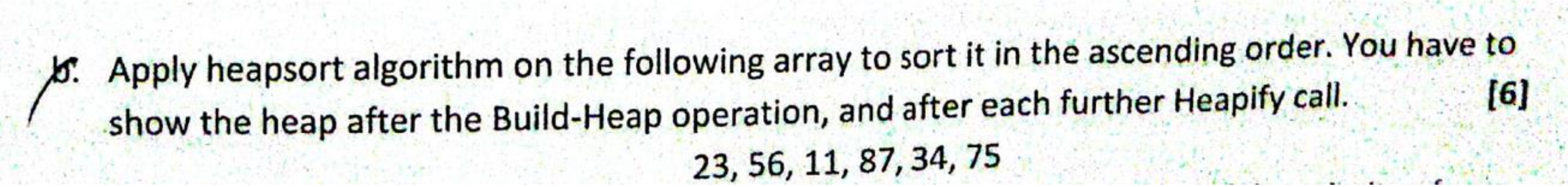
To sort an array in descending order using the heapsort algorithm with a max-heap, I **do not need to call the `Build-Heap` function before starting the sorting process**. Here's why:

In a max-heap, the largest element is already at the root (index 0). The heapsort algorithm works by repeatedly extracting the maximum element from the heap (which is the root in a max-heap) and placing it at the end of the array. After each extraction, the heap property is restored by calling `heapify` on the reduced heap. However, if the array is not a max-heap, I would need to call `Build-Heap` to transform it into a max-heap before starting the sorting process.

So, if the array is already a max-heap, I can proceed with the sorting without calling `Build-Heap`.

Heap, Heap Sort, Heapify

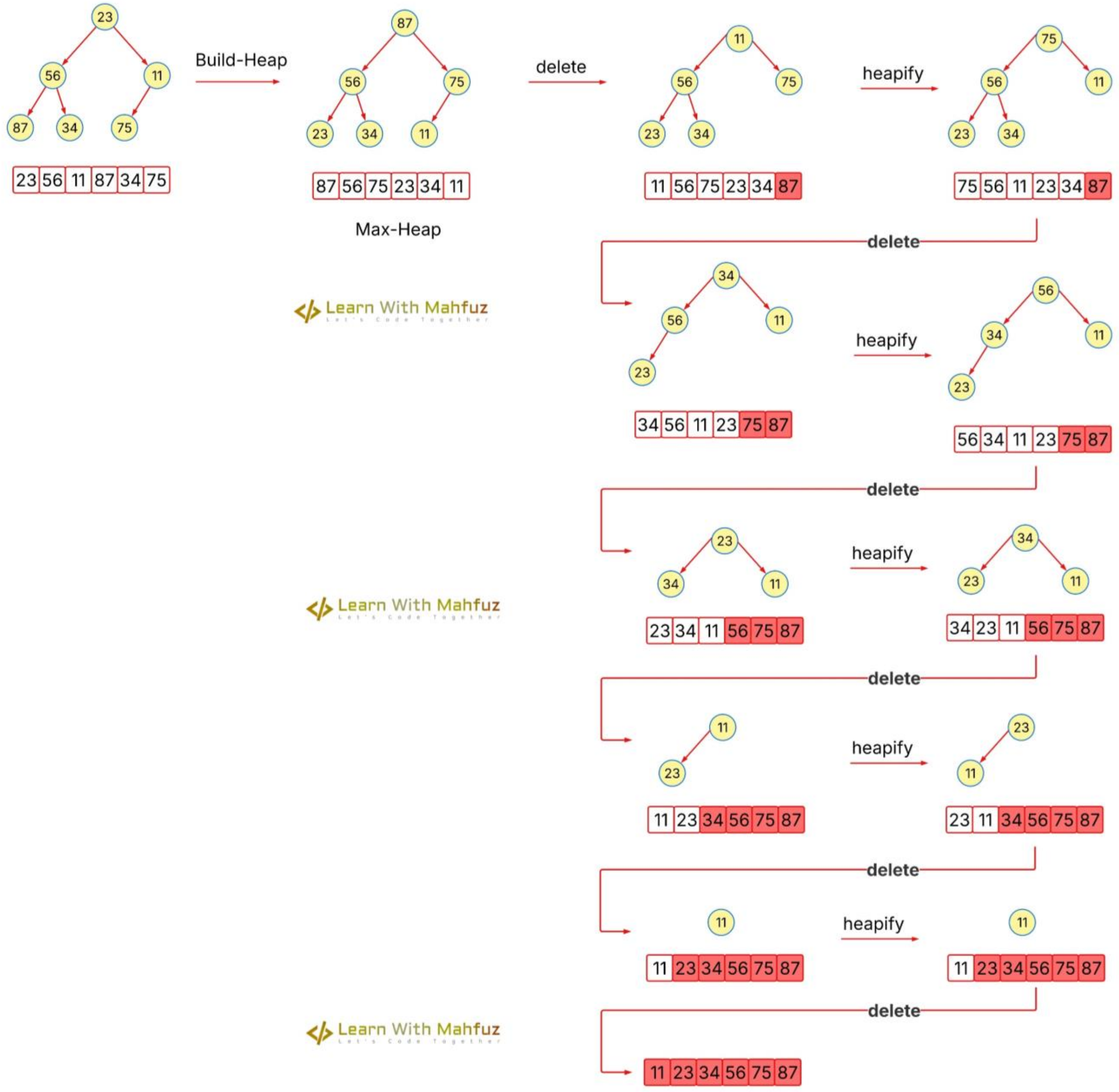
Summer - 24



b. Apply heapsort algorithm on the following array to sort it in the ascending order. You have to show the heap after the Build-Heap operation, and after each further Heapify call. [6]

23, 56, 11, 87, 34, 75

Summer - 24



23 56 11 87 34 75

87 56 75 23 34 11

11 56 75 23 34 87

75 56 11 23 34 87

34 56 11 23 75 87

56 34 11 23 75 87

23 34 11 56 75 87

34 23 11 56 75 87

11 23 34 56 75 87

23 11 34 56 75 87

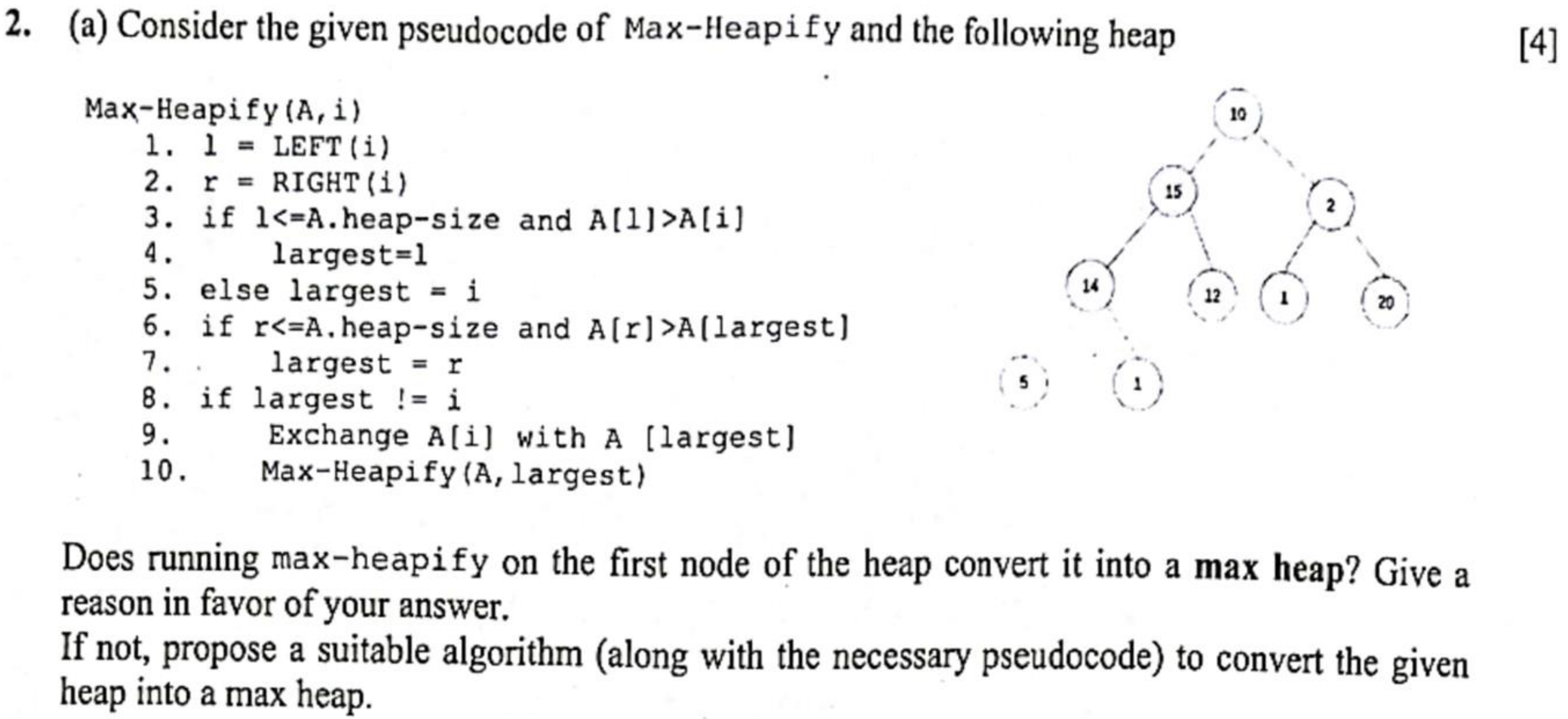
11 23 34 56 75 87

11 23 34 56 75 87

11 23 34 56 75 87

Heap, Heap Sort, Heapify

Summer - 23



2. (a) Consider the given pseudocode of `Max-Heapify` and the following heap [4]

```
Max-Heapify(A,i)
    l = LEFT(i)
    r = RIGHT(i)
    if l<=A.heap-size and A[l]>A[i]
        largest=l
    else largest = i
    if r<=A.heap-size and A[r]>A[largest]
        largest = r
    if largest != i
        Exchange A[i] with A [largest]
        Max-Heapify(A,largest)
```

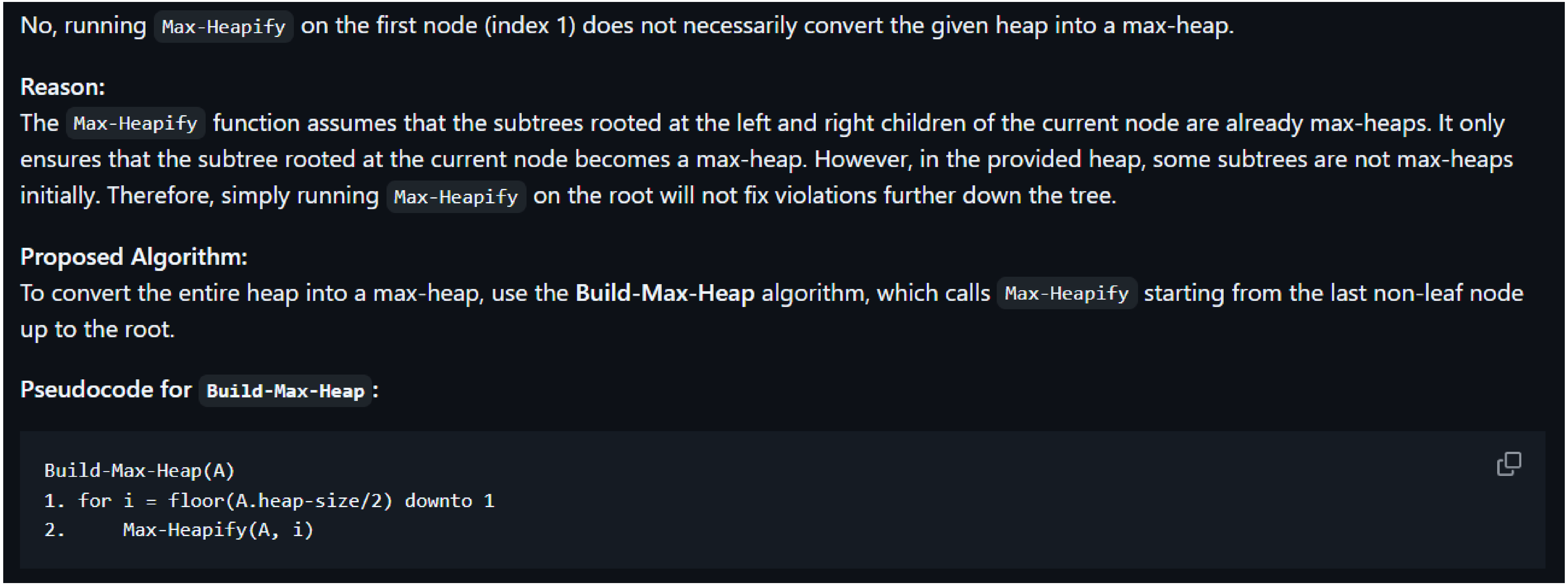
10
15
2
14
12
1
20
5
1

Does running `max-heapify` on the first node of the heap convert it into a **max heap**? Give a reason in favor of your answer.

If not, propose a suitable algorithm (along with the necessary pseudocode) to convert the given heap into a max heap.

Heap, Heap Sort, Heapify

Summer - 23



No, running `Max-Heapify` on the first node (index 1) does not necessarily convert the given heap into a max-heap.

Reason:
The `Max-Heapify` function assumes that the subtrees rooted at the left and right children of the current node are already max-heaps. It only ensures that the subtree rooted at the current node becomes a max-heap. However, in the provided heap, some subtrees are not max-heaps initially. Therefore, simply running `Max-Heapify` on the root will not fix violations further down the tree.

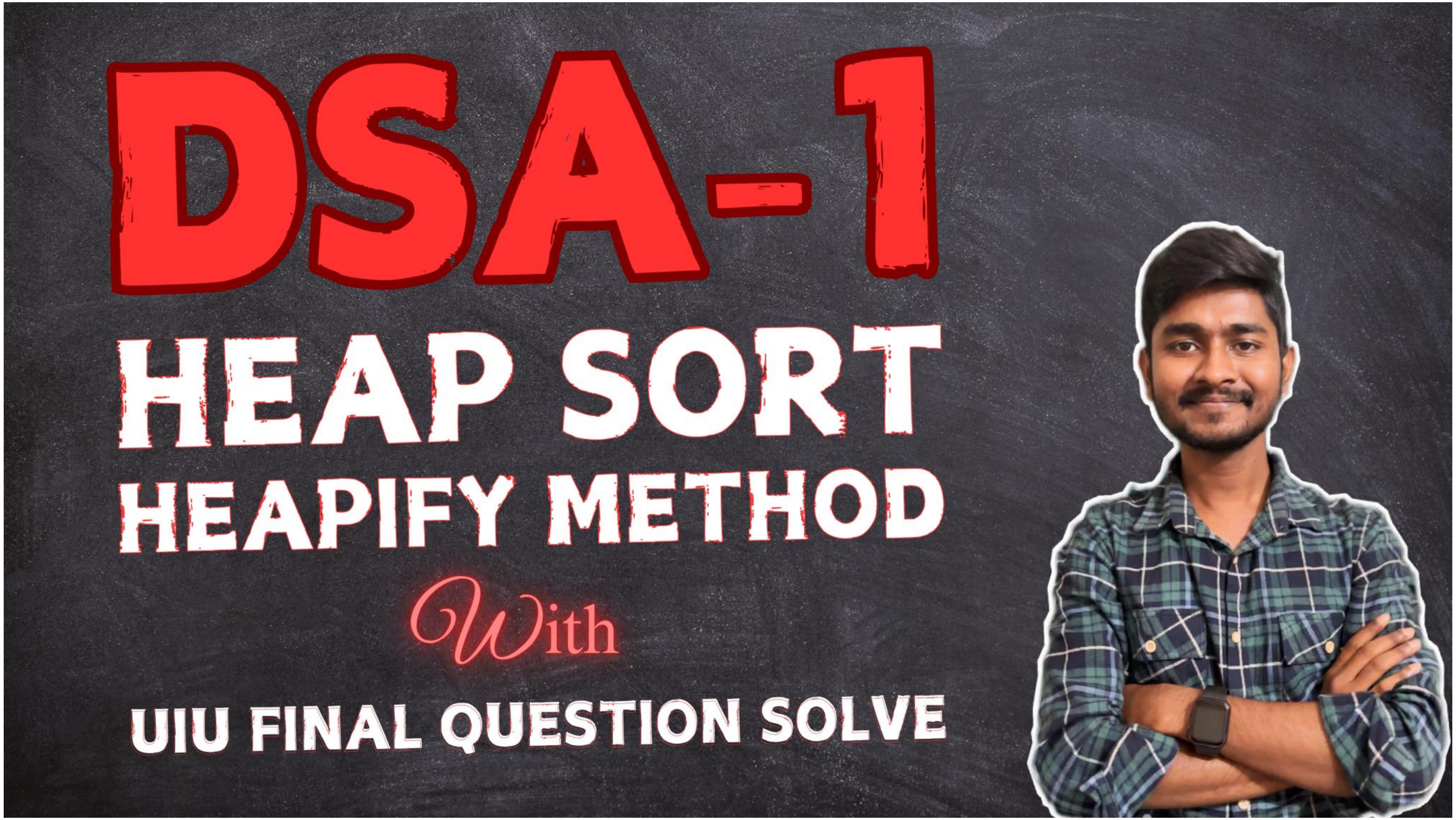
Proposed Algorithm:
To convert the entire heap into a max-heap, use the **Build-Max-Heap** algorithm, which calls `Max-Heapify` starting from the last non-leaf node up to the root.

Pseudocode for `Build-Max-Heap`:

```
Build-Max-Heap(A)
for i = floor(A.heap-size/2) downto 1
    Max-Heapify(A, i)
```

Click here to go to the **GitHub repository**

Learn With Mahfuz
Let's Code Together



Click here to see this video!

THANK YOU!

Learn With Mahfuz

Let's Code Together