CS2023 - Data Structures and Algorithms
In-class Lab Exercise

Week 8

Index: 200381U

You are required to answer the below questions and submit a PDF to the submission link provided under this week lab section before end of the session time (no extensions will be provided). You can either write / type your answers, but either way your answers should be readable.

Create GitHub repository, add your codes there and add respective link to the submission file.

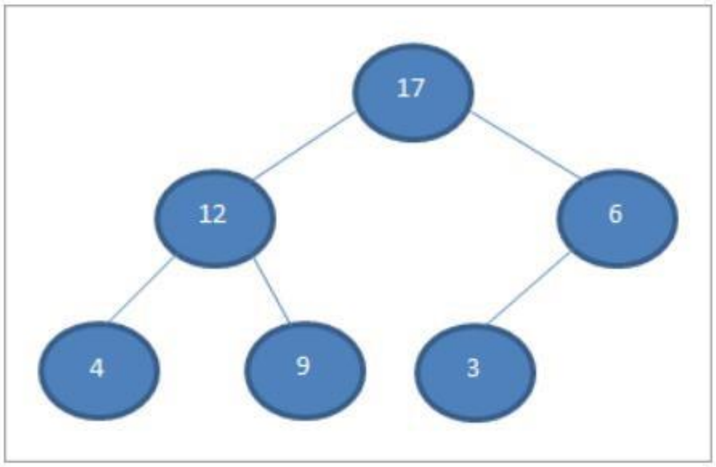
Exercise:

A binary heap is represented using a complete binary tree. A complete binary tree is a binary tree in which all the nodes at each level are completely filled except for the leaf nodes and the nodes are as far as left.

While representing a heap as an array, assuming the index starts at 0, the root element is stored at 0. In general, if a parent node is at the position i, then the left child node is at the position (2*i + 1) and the right node is at (2*i +2).

Sample Input array
4 17 3 12 9 6

Sample max-heap for above input



Implement Heapsort by creating max-heap in C++. Use the given "*heap.cpp*" file for your implementation.

- In the given program, the function heapify() is used to convert the elements into a heap using recursion.
- The function heapSort() sorts the array elements using heap sort. It starts from the nonleaf nodes and calls the heapify() on each of them. This converts the array into a binary max heap.
- The main() function only shows an example of a given set of numbers. Modify it to read numbers from the user or assign random values.

Sample Output:

```
Input array: 4
17 3 12 9 6
Sorted array:
3 4 6 9 12 17
```

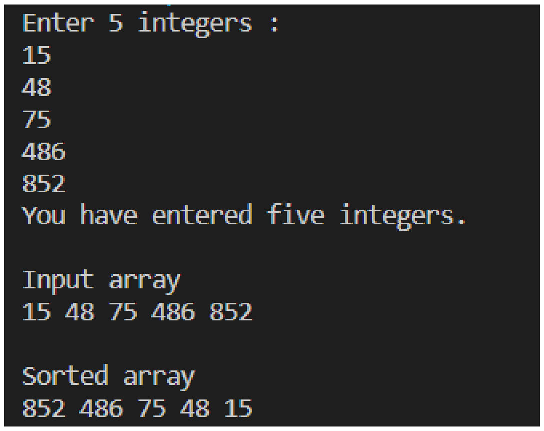
Submission:

- Include screen shots of terminal output, sample max-heap drawn for your input, and GitHub link for your implementation.
- Discuss on the time complexity of heap sort.
- Upload the answers as a PDF file named, "**In<no>_IndexNO_lab8.pdf**"

ANSWERS:

Terminal outputs:

Test 1:

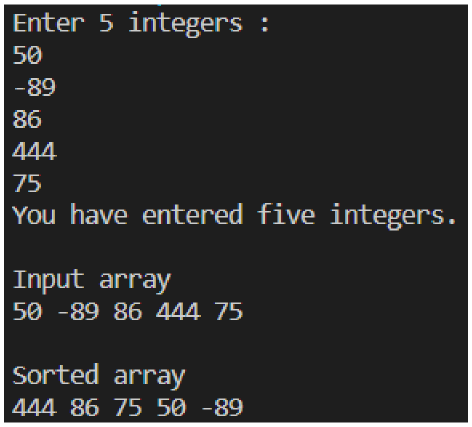


```
Enter 5 integers :
15
48
75
486
852
You have entered five integers.

Input array
15 48 75 486 852

Sorted array
852 486 75 48 15
```

Test2:

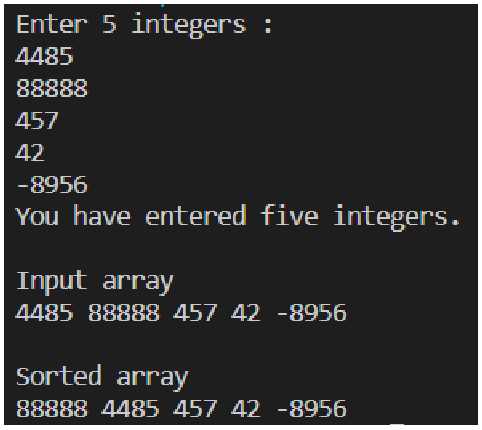


```
Enter 5 integers :
50
-89
86
444
75
You have entered five integers.

Input array
50 -89 86 444 75

Sorted array
444 86 75 50 -89
```

Test3:

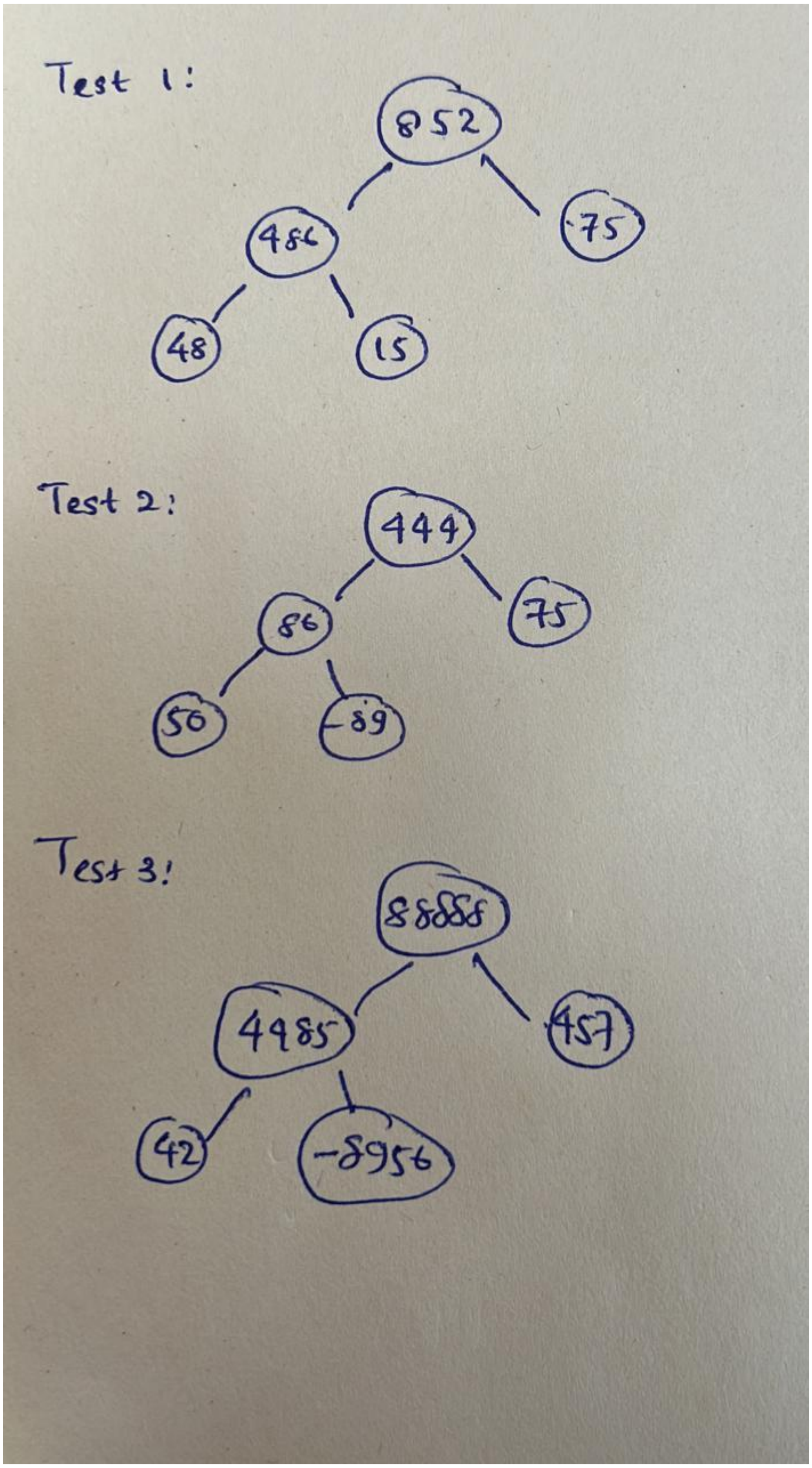


```
Enter 5 integers :
4485
88888
457
42
-8956
You have entered five integers.

Input array
4485 88888 457 42 -8956

Sorted array
88888 4485 457 42 -8956
```

Visualizing heaps:



Discussion on time complexity of heap sort:

Max heap sort is a variation of heap sort where the largest element is always at the root of the heap. In this variation, the time complexity remains the same as the general heap sort, which is O(n log n).

The algorithm works by first building a max heap from the array, which takes O(n) time. Then it repeatedly extracts the maximum element from the heap and places it at the end of the array, reducing the size of the heap by one. This process is repeated n times, resulting in a sorted array.

The extract-max operation takes O(log n) time since it involves reorganizing the heap to maintain the max heap property. This operation is performed n times in the worst case, resulting in a time complexity of O(n log n) for max heap sort.

Therefore, the time complexity of max heap sort is O(n log n), which is the same as the general heap sort.