CS2023 - Data Structures and Algorithms
In-class Lab Exercise

Week 8

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:

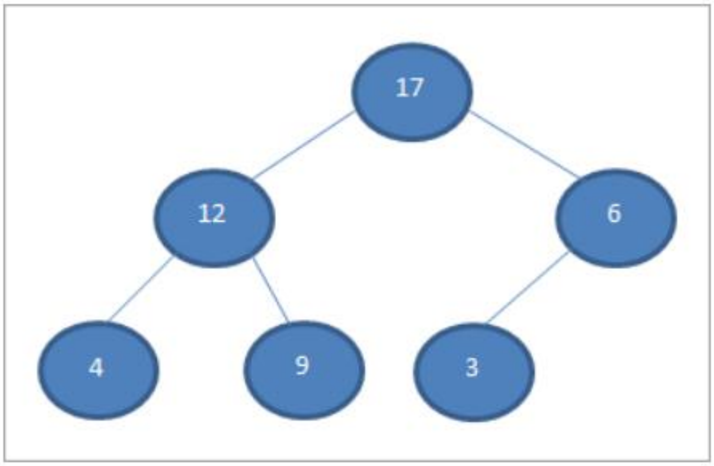
A binary heap is represented using a complete binary tree. A complete binary tree is a binary tree in which all the nodes at each level are completely filled except for the leaf nodes and the nodes are as far as left.

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 non-leaf 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**"