

# LAB - 5

## Heap & Priority Queues

(Building a heap, Heap sort algorithm, Min-Priority queue, Max-Priority queue)

### PROGRAM EXERCISE

#### Lab. Exercise (LE)

- 5.1** Write a menu (given as follows) driven program to sort an array of n integers in ascending order by heap sort algorithm and perform the operations on max heap. Determine the time required to sort the elements. Repeat the experiment for different values of n, the number of elements in the array to be sorted and plot a graph of the time taken versus n. The elements can be read from a file or can be generated using the random number generator.

#### **MAX-HEAP & PRIORITY QUEUE MENU**

0. Quit
1. n Random numbers=>Array
2. Display the Array
3. Sort the Array in Ascending Order by using Max-Heap Sort technique
4. Sort the Array in Descending Order by using any algorithm
5. Time Complexity to sort ascending of random data
6. Time Complexity to sort ascending of data already sorted in ascending order
7. Time Complexity to sort ascending of data already sorted in descending order
8. Time Complexity to sort ascending all Cases (Data Ascending, Data in Descending & Random Data) in Tabular form for values n=5000 to 50000, step=5000
9. Extract largest element
10. Replace value at a node with new value
11. Insert a new element
12. Delete an element

Enter your choice:

If the choice is option 8, the it will display the tabular form as follows:

#### **Analysis of Max-Heap Sort Algorithm**

Sl. No.	Value of n	Time Complexity (Sorted Data)	Time Complexity (Reversely Sorted Data)	Time Complexity (Random Data)
1	5000			
2	10000			

3	15000			
4	20000			
5	25000			
6	30000			
7	35000			
8	40000			
9	45000			
10	50000			

### Home Exercise (HE)

- 5.2** Similar to above program no.5.1, write a menu driven program to sort an array of n integers in **descending order by heap sort algorithm**. Hints: Use min heap and accordingly change the menu options.