National University of Computer and Emerging Sciences



# Lab Manual

*for*

# Data Structure

|  |  |
| --- | --- |
| Lab Instructor(s) | Anosha Khan |
| Semester | FALL 2024 |

Department of Computer Science FAST-NU, Lahore, Pakistan

**Lab Manual 09**

**Objectives:**

After performing this lab, students shall be able to revise:

* Minheap

## Problem 1

template<typename T>

class minHeap{

public:

minHeap(); // default constructor

minHeap(T\* arr, intN); // parameterizedconstructor that will take an array of random numbers and its size in parameters and initialize the heap with random values. It will call the buildMinHeap() function to convert the random values into a heap.

void buildMinHeap() // It will generate heap from random values stored in the object.

void insert(const T & x); // Inserts the key value in the heap array such that, the resultant heap tree is a complete binary tree and it follows min heap order.

bool isEmpty() const; // returns true if it is empty

const T &getMin() const; //returns minimum value this operation should be performed in O(1)

void deleteMin(); // deletes minimum value this operation should be performed in O(logN)

bool deleteAll(T key); //remove all occurrences of key value from the heap and update the heap accordingly.

private:

vector<T> \_vector;

void bubble\_up(int i); // A recursive method to heapify a subtree with the root at given index. It maintains heap property during insertion

void bubble\_down(int i); // It maintains heap property during deletion

};

Your task is to create a template-based minHeap class with the functions mentioned above and write main to perform the following tasks;

1. Insert following items in heap; 10, 40, 50, 5, 60, 15, 20
2. Find first,second and third minimum in min heap

int main()

{

int array[] = {10, 4, 5, 30, 3, 300};

minHeap obj(array, 6);

for(int i=0; i<3; ++i)

{

cout << obj.getMin()<< " ";

obj.deleteMin();

}

return 0;

}

1. Implement a non-member function Heapsort (T\* arr, intsize, int sorting\_order) that will take an array of random numbers, and its size in parameters from the user, along with the order of sorting 1 means ascending and 0 means descending. You may need to use the following functions to implement this.
2. Parameterized constructor
3. getMin()
4. deleteMin()