CMP 210 – Data Structures and Algorithms
Fall 2019
LAB-10

Issue Date: December 19, 2019

ALERT!

1. The objective of this lab is understanding min/max heap data structure.
2. This is an individual lab, you are strictly **NOT** allowed to discuss your solution with fellow colleagues, even not allowed asking how is he/she is doing, it may result in negative marking. You can **ONLY** discuss with your TAs or with me.
3. Beware of memory leaks and dangling pointers.

Task 01: [30 Marks]

Implement a Min Heap such that each node of this Min Heap will contain the roll number, name, and CGPA of a student. The heap will be organized on the basis of students' CGPAs i.e. the student having the minimum CGPA will be at the root of the heap. Use the array based implementation to construct min heap for those who want to recall that how the tree can be stored in and accessed from an array should see the paragraph in rectangle at the bottom of this page).

The class definitions will look like:

```
class Student
{
      friend class StudentMinHeap;
    private:
      string name;
      double cgpa;
      int rollNumber;

     // Methods…
};

class StudentMinHeap
{
      Student *arr;      //Array of students which, arranged like a Max Heap
      int curSize;      //Current number of students present in the heap
      int maxSize;      //Maximum number of students that can be present in heap

    public:
      StudentMinHeap (int size);      //Constructor
      ~StudentMinHeap();            //Destructor
      bool isEmpty();    //Checks whether the heap is empty or not
      bool isFull();    //Checks whether the heap is full or not
      StudentMinHeap& operator = (StudentMinHeap &); //Overloaded assignment operator

      // Methods…
};
```

A binary tree can be implemented using arrays such that each index of array holds a tree node. In array based implementation there may be following questions to be answered:

- What is the initial size of array as the tree starts growing from root to leaf node by node?
   You can take height of the tree as input and can find the maximum possible nodes of given height using $2^h – 1$. So you can create the array of maximum possible nodes of a given height and store the tree nodes in level order.
- Cool, but how to get children of a particular node?

On examining a tree and its array you may observe that all left children are on odd indices and all right children are on even indices. Thus, for each node at index i, the left child is at index $(2 \times i) + 1$, and the right child is at index $(2 \times i) + 2$. The parent of node i is at $\lfloor (i - 1)/2 \rfloor$.

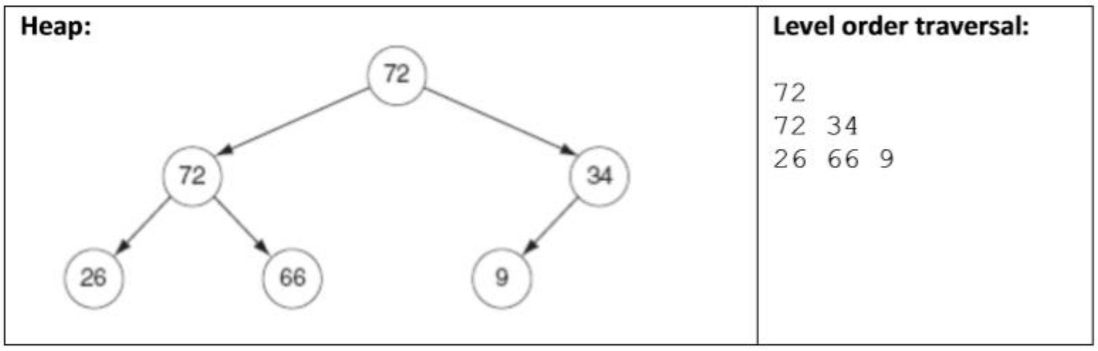
You are required to implement the following member functions of the StudentMinHeap class:

bool insert (int rollNo, string name, double cgpa)
This function will insert the record of a new student (with the given roll number, name and CGPA) in the Min Heap. This function should return true if the record was successfully inserted in the heap and it should return false otherwise, with worst case time complexity of O(lg n). If two students have the same CGPA then their records should be stored in a way such that the student with smaller roll number comes before the student with larger roll number.

Student extractLeastCGPA ()
This function will remove the record from the Min Heap which has the least CGPA. The worst case time complexity of this function should also be O(lg n).

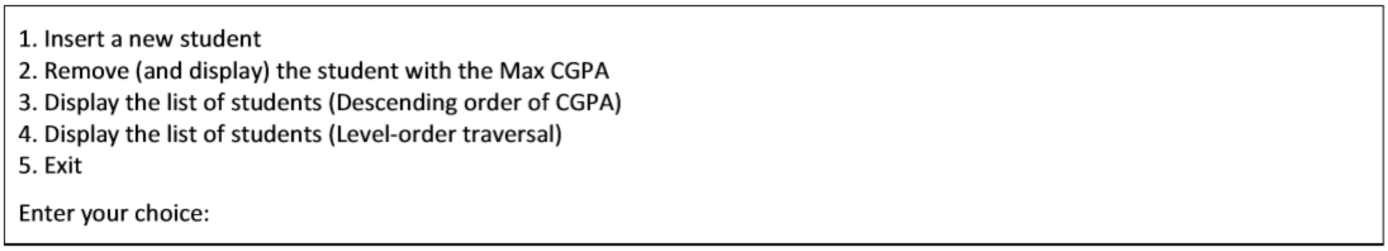
void displayStudentList()
This function will display the students list in ascending order of CGPAs. If two students have the same CGPA, then the student with smaller roll number will be displayed before the student with larger roll number. This function should not affect the heap i.e. after the execution of this function the heap should be left in the same way, as it was when this function was called.

void printHeap()
This function will perform a level order traversal of the StudentMinHeap and display the name, roll numbers and CGPAs of the students. The output displayed by the level order traversal should follow the format as shown below:



```
72
72 34
26 66 9
```

Write a menu-based driver function (menu) to illustrate the working of different functions of the StudentMinHeap class. The menu should look like:



```
1. Insert a new student
2. Remove (and display) the student with the Max CGPA
3. Display the list of students (Descending order of CGPA)
4. Display the list of students (Level-order traversal)
5. Exit

Enter your choice:
```