National University of Computer and Emerging Sciences

Laboratory Manual 14

for

Data Structures Lab

Department of Computer Science

FAST-NU, Lahore, Pakistan

Objectives:

In this lab, students will practice:

1. Implementation of Min Heap Using Arrays

Heap Data Structure

In this tutorial, you will learn what heap data structure is. Also, you will find working examples of heap operations in C++.

Heap data structure is a complete binary tree that satisfies the heap property, where any given node is

always greater than its child node/s and the key of the root node is the largest among all other nodes. This property is also called max heap property.

always smaller than the child node/s and the key of the root node is the smallest among all other nodes. This property is also called min heap property.

Please visit the given link for more understanding of Heap data structures.

https://www.geeksforgeeks.org/heap-data-structure/

Question 1:

a. Create a struct HeapItem as follows:

```
template <typename k, typename v>
struct HeapItem
{
	k key;
	v value;
};
```

b. Now create a MinHeap class which contains:

1. A pointer to HeapItem, "arr".
2. An int variable "capacity" which stores the total capacity of heap.
3. An int variable "totalItems" which contains the count of current total number of items stored.

Provide the following member functions for your MinHeap class:

1. A default constructor which assigns nullptr to arr pointer. `MinHeap()`

2. An overloaded constructor which takes as argument the value of capacity and allocates the memory of the required capacity to arr pointer. `MinHeap(int _capacity).`

3. An insert function which takes as argument a key value pair. It then inserts the key value pair in the heap array such that, the resultant heap tree is a complete binary tree and it follows min

heap ordering. If totalItems==capacity, then double the capacity of heap array and insert the key value pair. There must not be any memory leaks. `void insert(k key, v value)`

4. A getMin function which assigns the value of that HeapItem, whose key is minimum, to the parameter passed by reference. It does not delete that HeapItem from the heap. Use assert(totalItems>0) to throw an error if the heap is empty. `void getMin(v& _value)`

5. A deleteMin function which deletes the HeapItem which has the minimum key. The Heap must remain a complete binary tree and it must follow min heap ordering after deleteMin is called. User assert(totalItems>0) to throw an error if the heap is empty. `void deleteMin()`

6. A function isEmpty which returns true if the heap has no element. `bool isEmpty() const`

7. A destructor

Question 2:

a. Create a student class which contains rollNumber as int, name as string, and cgpa as float. Implement any constructors, getters and setters.

b. Overload the << operator to neatly print the student object on the screen.

c. Implement a function buildStudentHeap which takes as argument a file name and an initially empty min heap object by reference. Your task is to read student data from the file and insert Student objects into the min heap. The key will be roll number.
`void buildStudentHeap(string fileName, MinHeap<int, Student> &stdHeap)`

Question 3:

Run the following main program

```
int main()
{
   MinHeap<int, Student> stdHeap;
   buildStudentHeap("students.txt", stdHeap);

   while (!stdHeap.isEmpty())
   {
         Student s;
         stdHeap.getMin(s);
         cout << s <<endl<<endl;
         stdHeap.deleteMin();
   }
```

```
    system("pause");
}
```