

## EECS 560 Lab 7: Binary Min Heap

### Due date:

10/18/2020, 11:59 pm -Sunday

### Objective

Get familiar with basic Heap implementation with C++. Work on functions on Binary Min Heap

### Specification of the ADT

Refer to section 6.3 in Textbook. Implement the code for methods in figure 6.4 of Class interface for Binary heap. which has Type declaration of functions of Binary Heap. The implementation for some of these methods available as mentioned below.

1. Figure 6.8 Function to insert and element into heap with signature void insert( const Comparable & x )
2. Figure 6.12 Functions for DeleteMin with and without parameters passed . Corresponding signatures of functions are void deleteMin( ) and void deleteMin( Comparable & minItem )
3. Figure 6.12 Function percolateDown used to delete minimum element with signature void percolateDown( int hole )
4. Figure 6.14 Constructor with signature explicit BinaryHeap( const std::vector<Comparable> & items ) and Build Heap function with signature void buildHeap( )

### Requirements to completed from text from template:

5. Rename the Binary Heap class name into “myBinaryMinHeap”, instead of “BinaryHeap” as indicated in the textbook.
6. Implement explicit BinaryHeap( int capacity = 100 ) defined from textbook template. It was not implemented in textbook. Constructor with default behavior.
7. Implement public member function isEmpty with signature bool isEmpty( ) const. This verifies it heap is empty or not. defined from textbook template. It was not implemented in textbook.
8. Implement public member function findMin to return minimum element in the Heap with signature const Comparable & findMin( ) const defined from textbook template. It was not implemented in textbook.

9. Implement public member function insert to insert an element with rvalue as parameter with signature void insert( Comparable && x ) defined from textbook template.
10. Implement public member function make Empty with signature void makeEmpty( ) defined from textbook template to empty the heap. It was not implemented in textbook.

#### **Additional requirements:**

11. Implement a public function levelOrder. This should print all the elements of the heap in Level order into output file. This function should have the signature void levelOrder(std::ostream & out).
12. For UnderflowException{ } used in the implmenetation, **Modify this to use std::cerr to report the error details, instead of using UnderflowException.**

#### **Testing and Grading**

We will test your implementation using a tester main function, on a number of instances that are randomly generated. We will release the tester main function, several instances (will be different from the grading instances but will have the same format), and expected output together with the lab instruction via Blackboard. Your code will be compiled under Ubuntu 20.04 LTS using g++ version 9.3.0 (default) with C++11 standard.

The command line we are going to use for compiling your code is:

“g++ -std=c++11 main.cpp” (note that main.cpp will try to include the .hpp file you submit, and your .hpp file needs to be property implemented to compile successfully).

Your final score will be the percentage your program passes the grading instances. **Note that if your code does not compile (together with our tester main function), you will receive 0.** Therefore, it is very important that you ensure your implementation can be successfully compiled before submission.

#### **Submission and Deadline**

Please submit your implementation as two one .hpp file, with file name “myBinaryMinHeap\_[YourKUID].hpp”. For example, if my KU ID is c123z456, my submission will be files named “**myBinaryMinHeap\_c124z456.hpp**”. Submissions that do not comply with the naming specification will not be graded. Please submit through Blackboard.