CSCE 221 - Programming Assignment 4 Report **(20 points)**

Due April 14, 2021

First Name: Last Name: UIN: User Name: E-mail address:

Please list all sources in the table below including web pages which you used to solve or implement the current homework. If you fail to cite sources you can get a lower number of points or even zero, read more in the Aggie Honor System Office http://aggiehonor.tamu.edu/

Type of sources					
People					
Web pages (provide URL)					
Printed material					
Other Sources					

I certify that I have listed all the sources that I used to develop the solutions/code to the submitted work.

"*On my honor as an Aggie, I have neither given nor received any unauthorized help on this academic work.*"

Your Name (signature) Date

1. The description of the assignment problem.

2. The description of data structures and algorithms used to solve the problem.
   (a) Provide definitions of data structures by using Abstract Data Types (ADTs)

   (b) Write about the ADTs implementation in C++ (for all the three MPQs).

   (c) Describe algorithms used to solve the problem. For every MPQ (`UnsortedMPQ`, `SortedMPQ` and `BinaryHeapMPQ`), list the MPQ functions (`remove_min(), is_empty(), min(),` and `insert()`) and provide their descriptions.

   (d) Show the time complexity analysis for the following. Time complexity analysis means providing a **basic runtime function/recurrence relation**, **solution for recurrence relation with steps (wherever needed)** and a **Big-O** Notation:
      i. **Best, worst, and average case** of each of the MPQ functions (`remove_min(), is_empty(), min(), and insert()`) for `UnsortedMPQ.` (Note: Some functions may have same runtimes for all the cases. In that case, write the answer only once and mention that the runtime applies to all the cases.).

         A. Provide an **example for best, worst, and average case** for `UnsortedMPQ`.

      ii. **Best, worst, and average case** of each of the MPQ functions (`remove_min(), is_empty(), min(), and insert()`) for `SortedMPQ.` (Note: Some functions may have same runtimes for all the cases. In that case, write the answer only once and mention that the runtime applies to all the cases).

         A. Provide an **example for best, worst, and average case** for `SortedMPQ`.

      iii. **Best, worst, and average case** of each of the MPQ functions (`remove_min(), is_empty(), min(), and insert()`) for `BinaryHeapMPQ.` (Note: Some functions may have same runtimes for all the cases. In that case, write the answer only once and mention that the runtime applies to all the cases).

         A. Provide an **example for best, worst, and average case** for `BinaryHeapMPQ`.

3. A C++ organization and implementation of the problem solution

   (a) Provide a list and description of classes or interfaces used by a program such as classes used to implement the data structures or exceptions.

   (b) Provide features of the C++ programming paradigms like Inheritance or Polymorphism in case of object oriented programming, or Templates in the case of generic programming used in your implementation.

4. A user guide description how to navigate your program with the instructions how to:

   (a) compile the program: specify the directory and file names, etc.

   (b) run the program: specify the name of an executable file.

5. Specifications and description of input and output formats and files

   (a) The type of files: keyboard, text files, etc (if applicable).

   (b) A file input format: when a program requires a sequence of input items, specify the number of items per line or a line termination. Provide a sample of a required input format.

   (c) Discuss possible cases when your program could crash because of incorrect input (a wrong file name, strings instead of a number, or such cases when the program expects 10 items to read and it finds only 9.)

6. Provide types of exceptions and their purpose in your program (Answer only to the ones that are applicable for this assignment).

   (a) logical exceptions (such as deletion of an item from an empty container, etc.).

   (b) runtime exception (such as division by 0, etc.)

7. Include evidence of your testing by providing screenshots. Screenshots should show execution of the 5 main methods (`unsortedmpq-main.cpp, sortedmpq-main.cpp, main.cpp, cpu-job-main.cpp, binaryheap-mpq-main.cpp`).

8. Provide graphs and data tables of your CPU timing simulation results. Graph should be plotted for **runtime vs. input size**. The input sizes are 4, 10, 100, and 1,000. To obtain this data, compile and run `main.cpp`. Choose option "2. Timing Simulation". Provide the input file name (SetSize4.txt) and output filename. After execution, you will find the output file in "OutputFiles" folder. The timing for all the three MPQ implementations will be displayed. Fill it in the following table and plot it as a **graph**.

	Runtime		
Input Sizes	Unsorted MPQ	Sorted MPQ	Binary heap MPQ
4 (SetSize4.txt)			
10 (SetSize10.txt)			
100 (SetSize100.txt)			
1000 (SetSize1000.txt)			