## CS 201, Fall 2022

## Homework Bonus

**DUE: January 9, Monday @23:59**

**Please check the submission rules towards the end of the document.**

**Points will be deducted in case of a violation of these rules!**

**Description:** In this assignment, you will write a C++ program that can **compress/transform a text file and reconstruct (decompress/inverse transform) a compressed/transformed text by using Burrows Wheeler Transform (BWT)**. The goal of the assignment is to gain experience in implementation of efficient algorithms and data structures in C++ for developing a practical tool.

**1) The Usage of the Program**

The program will be executed from the command prompt.

Assume that the executable is named as "bwt.exe", and there exists a text file, called "data.txt", in the same directory as the program executable.

Then, the following command should parse the text in "data.txt", generate a new file (or overwrite an existing one) called "data.bwt", which includes the corresponding compressed text.

bwt c data.txt data.bwt

Then, the following command should decompress "data.bwt" and put the reconstructed text in "data2.txt"

bwt d data.bwt data2.txt

The contents of the files "data.txt" and "data2.txt" should be the same.

Notice that we use the same executable ("bwt.exe") for two functions (compresss/decompress) with command line arguments (c/d). The name of the input and output files are also obtained as command line arguments.

The program should be **robust** against usage errors. If the provided arguments are wrong/missing, or if the input file cannot be found, the program should print out an informative error message.

Short tutorials on command line arguments and file I/O in C++ can be found in the following links (many more resources can be found on the Internet):

* *Parsing command line arguments:*

http://www.site.uottawa.ca/~lucia/courses/2131-05/labs/Lab3/CommandLineArguments.html

* *Reading from / writing to files in C++:*

http://www.cs.hmc.edu/~geoff/classes/hmc.cs070.200109/notes/io.html

http://courses.cs.vt.edu/~cs2604/fall02/binio.html

**2) The Implementation**

**BWT method** will be used for the implementation of the text compression algorithm. The method organizes a character string into runs of similar characters. Such reorganization is important for compressing the text afterwards. This is useful for compression, since it tends to be easy to compress a string that has runs of repeated characters. BWT transform is reversible, which allows for lossless compression.

A comprehensive description of the **BWT method** can be found in the following links:

https://medium.com/@mr-easy/burrows-wheeler-transform-d475e0aacad6

<https://www.cs.tufts.edu/~ablumer/BWTSlides.pdf>

**In this assignment, you cannot use the existing sort implementations in C++ libraries. You need to implement a sorting algorithm from scratch, it can be any of the sorting algorithms we discussed in the class.**

**WARNING:** You can benefit from sources on the Internet to learn more about the BWT method, implementation variations and practical issues (e.g., how to read command line arguments in C++). However, the submitted implementation must be of your own.

**3) Submission**

Youwill submit this homework via the LMS system. You should follow the file-naming conventions and guidelines below.

* You should submit your source files as a **ZIP** archive file (**NOT** RAR or other formats). The name of the file should be in format “**<USER-ID>\_hwbonus.zip**”. For example, if your username is vy1043, then the name of the submitted file should be “vy1043\_hwbonus.zip”. Pay attention that all the letters are in lower-case. ZIP archive is supposed to contain **just the source files**, no folders are allowed by any means.
* The contents of the ZIP file should be **bwt.cpp** (includes the *main* function) and optionally any other class definition and implementation files depending on the data structures you utilize.
* Late submissions and C++ files that do not compile are **not** accepted.
* You can resubmit your homework (until the deadline) if you need to.
* Make sure that your program does **not** include commands specific to a development environment, e.g., *system(“pause”)* or *#pragma once* in Visual Studio.