HW4 – String Compression

CSS 501 – Data Structures & Object-Oriented Programming
By: Hansel Ong

Summary

The goal of compression algorithms[1] are to represent the same exact data in a manner that takes up less space to store (e.g. by somehow reducing redundancies in the data). Let's experiment with a simple string compression problem.

Skills Expected

- All previous skills up to now
- STL

Assignment Description

Design and create a "CompressedString" class where it takes a String in the constructor and stores it internally as a "compressed" vector of chars. Create an << overload that will return the "de-compressed" string.

For example, given the following string:
aaabbeeee it should compress it to → { '3', 'a', '2', 'b', '4', 'e' }

Grading Criteria

- "CompressedString"
  - [2 Points] Constructor as defined
  - [2 Points] Use of vector
  - [3 Points] Compression algorithm/method
    - [1 Point] Print to console (cout) how much shorter the "string" is after compression, e.g. in the above example, we went from 9 characters to 6 characters, so the string is compressed by 33%
  - [2 Points] overload<< that returns the de-compressed string
- Demonstration
  - [1 Point] Compression of a short string, e.g. "abc"
  - [1 Point] Compression of a long string, e.g. "aaaaaddddddeeeeffwwwwazwww"
  - [1 Point] Printing out the de-compressed string (cout)
- [2 Points] Big-O Analysis of the "Compress" and "Decompress" function

[1] https://en.wikipedia.org/wiki/Data_compression