ISSN: 0974-6471, Vol. 10, No. (3) 2017, Pg. 703-707

Oriental Journal of Computer Science and Technology

Journal Website: www.computerscijournal.org

# Performance Analysis of Data Compression using Lossless Run Length Encoding

**CHETAN R. DUDHAGARA**[1] **and HASAMUKH B. PATEL**[2]

[1]Computer Science Department N. V. Patel College of Pure and Applied Sciences
Vallabh Vidyanagar, Anand, Gujarat, India.
[2]Computer Science Department N. V. Patel College of Pure and Applied Sciences
Vallabh Vidyanagar, Anand, Gujarat, India.

**Abstract**

In a recent era of modern technology, there are many problems for storage, retrieval and transmission of data. Data compression is necessary due to rapid growth of digital media and the subsequent need for reduce storage size and transmit the data in an effective and efficient manner over the networks. It reduces the transmission traffic on internet also. Data compression try to reduce the number of bits required to store digitally. The various data and image compression algorithms are widely use to reduce the original data bits into lesser number of bits. Lossless data and image compression is a special class of data compression. This algorithm involves in reducing numbers of bits by identifying and eliminating statistical data redundancy in input data. It is very simple and effective method. It provides good lossless compression of input data. This is useful on data that contains many consecutive runs of the same values. This paper presents the implementation of Run Length Encoding for data compression.

**Article History**

Received: 17 July 2017
Accepted:09 August 2017

**Keywords**

Technology,
Transmission,
Compression,
Redundancy,
Run Length Encoding.

## Introduction

Data compression methods will compress the original file such as text, image, audio or video into different file which is called compressed file. It is a technique used for decreasing the requirements of storage space in different storage media. It also requires less bandwidth for data transmission over the network. It also reduces the time for displaying and loading data.

Data compression is possible because mostly all data available in the word may contain redundancy. The data compression method is used to for converting data to easy to use format. Similarly, decompression method returns the data to its original form. Run Length Encoding (RLE) is one of the most significant and widely used encoding techniques for compressing any type of data. This method is work on repetition of characters in inputted string of data.

**CONTACT** DR. CHETAN R. DUDHAGARA ✉ crdudhagara32@gmail.com Computer Science Department N. V. Patel College of Pure and Applied Sciences Vallabh Vidyanagar, Anand, Gujarat, India.

To link to this article: http://dx.doi.org/10.13005/ojcst/10.03.22

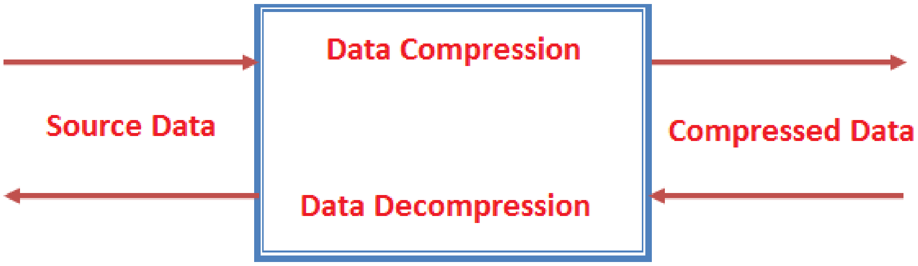


**Fig. 1 : Data Compression and Decompression**

**Types Of Data Compression**

The data compression methods are mainly classified into two different categories based on reconstruction of original data from the compressed data. The categories are as follows:

- Lossless Data Compression
- Lossy Data Compression

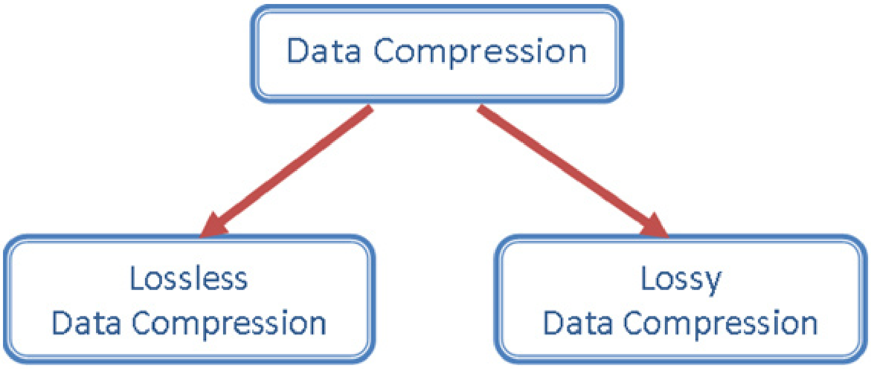
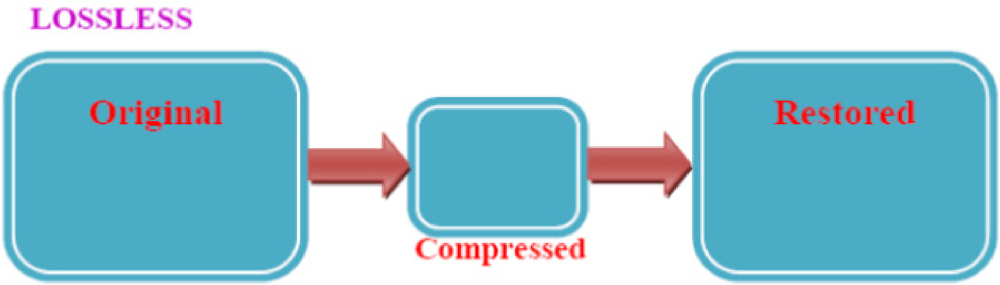


**Fig. 2 : Classification of Data Compression**

**Lossless Data Compression**

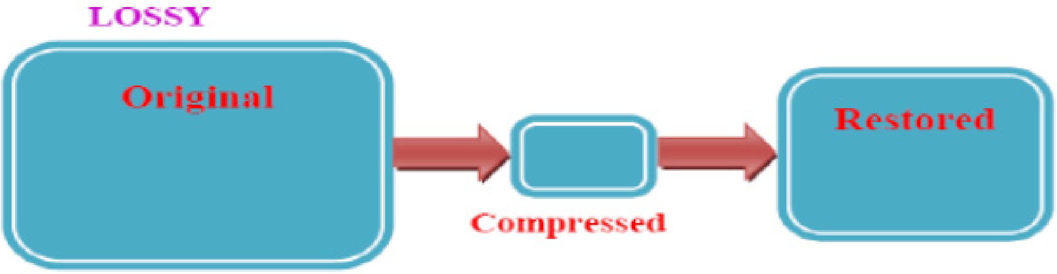
In lossless data compression methods, the size of original data will reduce and store into compressed file. That compressed file may transfer over networks. We can regenerate the same original file from the compressed file by using data decompression methods.

The main feature of lossless techniques is we get same data or file from decompressed file. Both original and reconstructed file are same size and quality. This algorithm usually exploits statistical redundancy. Generally, every real word data may contain statistical redundancy, so lossless data compression is possible.



**Lossy Data Compression**

Lossy data compression is technique in which regenerated data or file from the decompression process may be same or not exactly same. The quality of data or image may be reducing with compare to original data. After applying the lossy techniques to the data, the data of file may not be reconstructed same as original file. This compression method is not good for critical data such as text data.



**Run Length Encoding**

RLE is s very simple compression techniques. It generally works on string of data. It is mostly used in repetitive patterns are available in data. This technique mainly works on reduction of repeating character from the inputted string. The repeat characters in the string are known as run. It occupies only two bytes in size. One byte requires for display occurrence of number of characters in run which is known as *run count*. The range of run will be 0 to 127 or 0 to 255 characters. The second byte requires for display the repeated character in the run. The range of this is 0 to 255 characters. It is known as *run value*.

**Case – I**

In the below string, the character P run of 8 P character would require 8 bytes to store.

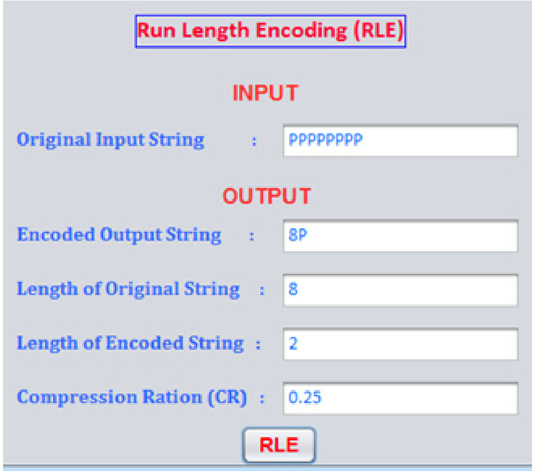
PPPPPPPP

The same string requires only 2 bytes after RLE encoding.

8P

The code which is produce from the string of character is known as RLE *packet*. In our case the code 8P is known as RLE *packet*.

In this case, the first byte, 8 is known as *run count,* which represents the number of occurrence of specific character. The second byte, P is known as *run value,* which represents the repeated character.



Run Length Encoding (RLE)

INPUT

Original Input String : PPPPPPPP

OUTPUT

Encoded Output String : 8P

Length of Original String : 8

Length of Encoded String : 2

Compression Ration (CR) : 0.25

**Case – II**

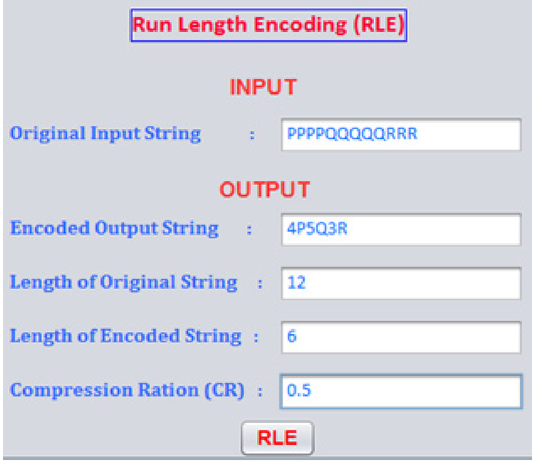
The below string contains the 12 character by using three different character runs.

PPPPQQQQQRRR

By using RLE encoding this can be compressed into three 2-bytes packets.

4P5Q3R

So, in this case after use of RLE on inputted string 12 byte string occupies six bytes only. In this case compression ratio is 2 : 1.



Run Length Encoding (RLE)

INPUT

Original Input String : PPPPQQQQQRRR

OUTPUT

Encoded Output String : 4P5Q3R

Length of Original String : 12

Length of Encoded String : 6

Compression Ration (CR) : 0.5

**Case – III**

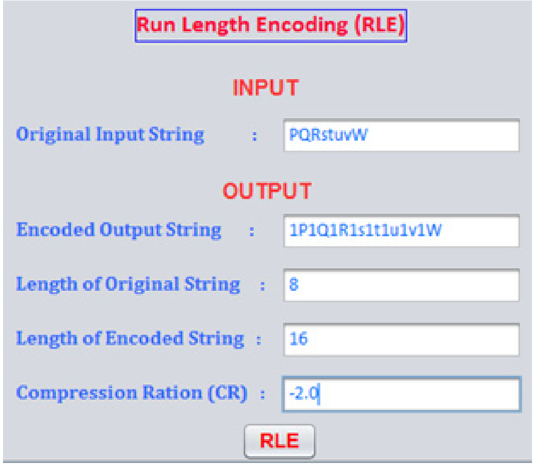
The below string contains the 8 character by using eight different character runs.

PQRstuvW

By using RLE encoding it require 16 bytes.

1P1Q1R1s1t1u1v1W

To implement RLE on long string, we require at least two characters. So that one single character occupies double bytes for storage.



Run Length Encoding (RLE)

INPUT

Original Input String : PQRstuvW

OUTPUT

Encoded Output String : 1P1Q1R1s1t1u1v1W

Length of Original String : 8

Length of Encoded String : 16

Compression Ration (CR) : -2.0

It is a fast and very simple technique for compression of data. The compression ratio is calculated on bases of inputted data only.

**Problem with RLE**

When we get 1 as a run length, then after implementation of RLE it requires two bytes. So that RLE require more bytes in the case where 1 as run length of data.

Consider the case – III, it require double size of storage space. It is a disadvantage of use of RLE.

**Conclusion**

There are two categories for data compression: Lossless and Lossy. There are many lossless data compression algorithms are available. In this paper we have present the lossless run length encoding techniques for data compression. Three different cases are discussed in this paper. Case – I gives the high compression ratio, case – II gives comparatively low compression ratio but case – III gives negative compression ratio.

So, it conclude that the in RLE, the compression ratio is totally depends on inputted data. If data contains redundancy then it gives high compression ratio.

## References

1 Hiroj Naik, "A Review on Image Compression using Different Techniques", *International Journal of Advanced Research in Computer Science and Software Engineering*, Volume **5**, Issue 2, February 2015.

2 N. S. Sethi, H. Singh, Amarjit Kaur, "A Review on Data Compression Techniques", *International Journal of Advanced Research in Computer Science and Software Engineering*, Volume **5**, Issue 1, February 2015.

3 S. Porwal, Y. Chaudhary, J. Joshi and M. Jain, "Data Compression Methodologies for Lossless Data and Comparison between Algorithms", I*nternational Journal of Engineering Science and Innovative Technology (IJESIT)* Volume **2**, Issue 2, March 2013

4 S. Shanmugasundaram and R. Lourdusamy, "A Comparative Study of Text Compression Algorithms" *International Journal of Wisdom Based Computing*, Vol. **1** (3), December 2011.

5 S. R. Kodituwakku and U. S. Amarasinghe, "Comparison of Lossless Data Compression Algorithms for Text Data", *Indian Journal of Computer Science and Engineering* Vol. **1** No. 4 Page No. 416-425

6 R. Kaur and M. Goyal, "An Algorithm for Lossless Text Data Compression" *International Journal of Engineering Research & Technology (IJERT)*, Vol. **2** Issue 7, July - 2013

7 H. Altarawneh and M. Altarawneh, "Data Compression Techniques on Text Files: A Comparison Study", *International Journal of Computer Applications*, Volume **26** No.5, July 2011.

8 K. Rastogi, K. Sengar, "Analysis and Performance Comparison of Lossless Compression Techniques for Text Data", *International Journal of Engineering Technology and Computer Research (IJETCR)* **2** (1) 2014, 16-19.

9 Amandeep Singh Sidhu, Er. Meenakshi Garg, "Research Paper on Text Data Compression Algorithm using Hybrid Approach", *International Journal of Computer Science and Mobile Computing,* Vol. **3,** Issue. 12, December 2014, page no. 01 – 10.