

Abstract

Data compression is a process changing several data to become certain code for saving storage and data transmission time. There are many algorithm of compression, one of them is LZW (Lempel-Ziv-Welch)[1].

Basically, LZW is the text file compression algorithm using array based dictionary method. LZW has weaknesses in string entering and string searching in its brute force dictionary, so possibly it can be optimized with other data structure for handling its dictionary. It uses a binary search tree.

Binary search tree data structure that has different characteristics with array data structure shall have impact when it is used as LZW dictionary; that is the bigger compression results ratio comparing than LZW array based dictionary. It is caused by difference way code words dictionary saving. Binary search tree data structure needs more information storages such as right child, left child, parent, and identity (id) from each node where this condition is different with array that only saves index information from each code word.

This final task purpose is for knowing factors that has effects to ratio, compression speed, and decompression speed of LZW algorithm with binary search tree as dictionary.

Key word: compression, decompression, brute force, ratio, array based dictionary, binary search tree.