

Data Structures Lab

Sheet 7

In this assignment, you will be implementing the famous Lempel-Ziv-Welch (LZW) compression algorithm. Most compression algorithms including UNIX compress are based on the LZW algorithm. The LZW algorithm compresses a given text by figuring out patterns which appear very often in the text. The basic idea is to replace (perhaps very long) patterns which appear many times by a much shorter code. A good description of the algorithm can be found here (<https://sites.google.com/site/balaiitr/LZW2.pdf>), along with a working example of how to compress and decompress strings. Your assignment is to implement the LZW algorithm. You can assume that the input text contains English characters (capital and small), space, and full stop. To implement the dictionary of code and key, you will use a hash table. Use your own hash function, but you must use quadratic probing to resolve collisions.

Specifics:

1. You must not re-use the dictionary from compression phase during the decompression. The dictionary must be re-built.
2. Initially (during both encoding and decoding), the dictionary contains the 256 possible ASCII characters. These are given codes 0-255 (ASCII encoding -- this means 'A' is given the code 65, and 'a' the code 97). You can start with an initial hash table size of 275 entries ($275-256=19$ entries are initially unused).
3. Use your own hash function.
4. Collisions must be resolved using quadratic probing.
5. The hash table must be growable. You may encounter scenarios where the hash table fills up -- these must be handled by suitably increasing the size of the hash table.
6. You may use any of the structures in the Standard Template Library, other than the hash table.

Deliverables:

Two programs -- a compressor and a decompressor -- written in JAVA or C++ is expected.

1. Given an input file as command line argument, the compressor must create a compressed file in the same directory.
2. Given a compressed file as a command line argument, the decompressor must create the uncompressed file in the same directory.

Example:

Input String = BABAABAAA