

# Report - assignment 8

Camilla K. Birkelund

IDATT2101

November 8th, 2022

## Assignment description

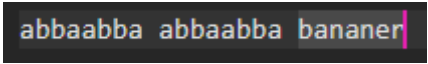
The assignment was to create a program that could compress and decompress a file using the Lempel-Ziv and Huffman algorithms. The requirements for the solution was that the resulting decompressed file would have to be equal to the original file, and the compressed file had to be smaller than the original file.

## Implementation

The source code can be found in “LZ77.java”, “Huffman.java” and “client.java”. To run the program, the LZ77.java, Huffman.java and client.java files have to be open. When running client.java, the user gets the option to either compress or decompress a file (or quit the program). The program then asks which file to perform the operations on. The default method to compress is to first use Lempel-Ziv 77, and then Huffman. The result will be written to a filename of the user's choice during the program running. With decompression, the program performs Huffman first, then Lempel-Ziv.

## Testing

During the implementation, several tests were performed on a small .txt file to make sure that the algorithms for compression and decompression gave the expected results.

The image shows a dark-themed text editor window. Inside, the text 'abbaabba abbaabba bananer' is displayed. The first two words are in a light blue color, and the third word 'bananer' is in a light red color. A vertical cursor is positioned at the end of the text.

The original file.

## Testing the Lempel-Ziv 77 algorithm

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded Text
00000000	00	00	00	09	61	62	62	61	61	62	61	20	FF	FF	FF		a b b a a b b a . . .
00000010	F7	00	00	00	09	00	00	00	07	62	61	6E	61	6E	65	72	. . . . . b a n a n e r

The compressed file.

```
≡ decompst
1 |abbaabba abbaabba bananer
```

The decompressed file.

## Testing the Huffman algorithm

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded Text
00000000	00	00	00	06	00	00	00	01	65	00	00	00	02	20	00	00	. . . . . e . . . . .
00000010	00	01	72	00	00	00	0A	61	00	00	00	02	6E	00	00	00	. . . r . . . . . a . . . . . n . . .
00000020	09	62	00	00	00	34	79	E8	F3	D3	52	AD	70				. b . . . . . 4 y . . . . . R . p

The compressed file

```
≡ decomp
1 |abbaabba abbaabba bananer
```

The decompressed file.

## Testing the Lempel-Ziv and Huffman together

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded Text
00000000	00	00	00	0B	00	00	00	01	20	00	00	00	01	07	00	00	. . . . . . . . . . .
00000010	00	01	F7	00	00	00	02	09	00	00	00	01	65	00	00	00	. . . . . . . . . . . e . . . . .
00000020	06	61	00	00	00	03	FF	00	00	00	09	00	00	00	00	02	. . . a . . . . . . . . . . .
00000030	6E	00	00	00	05	62	00	00	00	01	72	00	00	00	61	A9	n . . . . . b . . . . . r . . . . . a .
00000040	8D	83	63	89	2F	54	D5	7D	87	1F	FE	80					. . . c . . . / T . . . . . }

The compressed file.

```
≡ decompressed
1 |abbaabba abbaabba bananer
```

The decompressed file.

# Results

The requirement from the assignment description was to be able to compress and decompress at least two of the attached files, which are much larger than the testfile used to check the algorithms. The files used in this case “diverse.txt”, “diverse.lyx” and “diverse.pdf”.

The first requirement from the program is that the size of the compressed file has to be smaller than the original. The compressed files were smaller than the original files in all three cases. When compressing “diverse.txt”, which is 16764 bytes, the resulting file is 11134 bytes. When compressing “diverse.lyx” the resulting file is 39121 bytes, whereas the original is 181076 bytes. The “diverse.pdf” file was originally 156671 bytes, and the compressed version was 147789. The difference between the compressed file and the original file for “diverse.pdf” might have been bigger if the file was compressed with only Lempel-Ziv, because the Huffman algorithm may have expanded the file size due to the Huffman tree structure.

The second requirement was that the decompressed file is equal to the original file. When testing the compressed, then decompressed version of diverse.txt, diverse.lyx and diverse.pdf against the original diverse.txt, diverse.lyx and diverse.pdf files, there are no differences.

```
diff diverse.txt diversedcomp
```

The result of checking the differences between the original file and the compressed and decompressed version of diverse.txt.

```
diff diverse.lyx lyxdecomp
```

The result of checking the differences between the original file and the compressed and decompressed version of diverse.lyx.

```
diff diverse.pdf pdfdecomp
```

The result of checking the differences between the original file and the compressed and decompressed version of diverse.pdf