



Brent Van Wynsberge

3^e bachelor Informatica, Universiteit Gent

Algoritmen en Datastructuren III

Stamnummer: 01201853

5 december 2017

Huffman compressie

Project Algoritmen en Datastructuren III

Inhoudsopgave

1	Algoritmen	1
1.1	Statische huffman	1
1.1.1	Algemene operaties	1
1.1.2	Encoderen	2
1.1.3	Decoderen	3
1.2	Adaptive huffman	3
1.2.1	Algemene operaties	3
1.2.2	Encoderen	3
1.2.3	Decoderen	3
1.3	Adaptive huffman met sliding window	3
1.3.1	Algemene operaties	3
1.3.2	Encoderen	3
1.3.3	Decoderen	3
1.4	Two pass adaptive huffman	3
1.4.1	Algemene operaties	3
1.4.2	Encoderen	3
1.4.3	Decoderen	3
1.5	Bloksgewijze adaptive huffman	3

1.5.1	Algemene operaties	3
1.5.2	Encoderen	3
1.5.3	Decoderen	3
2	Datasets	4
3	Experimenten	5
4	Besluit	6
4.1	Statische huffman	6
4.2	Adaptive huffman	6
4.3	Adaptive huffman met sliding window	6
4.4	Two pass adaptive huffman	6
4.5	Bloksgewijze adaptive huffman	6

1. Algoritmen

In dit onderdeel bekijken we de pseudocode van alle geïmplementeerde algoritmen. Wanneer er een niet triviale operatie opgeroepen wordt, wordt deze beschreven in het onderdeel 'Algemene Operaties'.

De `WRITE(foo)` operatie heeft als betekenis: schrijf `foo` weg naar `stdout`. De operatie abstraheert het eventuele gebruik van buffers of de alignering van de bits weg.

De `READ` operatie heeft als betekenis: lees de volgende byte van `stdin`. Analooq met `WRITE` abstraheert dit het gebruik van buffers weg en doet dit alsof de tekst altijd byte per byte wordt ingelezen.

1.1 Statische huffman

1.1.1 Algemene operaties

Algoritme 1: `incrementWeight`

1 *input*

Algoritme 2: `readInput`

```
1 while (byte  $\leftarrow$  READ( ))  $\neq$  EOF do
2   |   input.content[input.size++]  $\leftarrow$  byte
3   |   INCREMENTWEIGHT(input, byte)
4 end
5
```

1.1.2 Encoderen

Algoritme 3: encode

```
1 input ← READINPUT( )
2 tree, max ← BUILDTREE(input)
3 codes ← BUILDDICTIONARY(tree, max)
4 PRINTTREE(tree)
5 ENCODEINPUT(input, tree)
6 zeros ← 0
7 while not FULL(last_byte) do
8   | last_byte[current_bit++] ← 0
9   | zeros ++
10 end
11
12 WRITE(last_byte)
13 WRITE(zeros)
```

1.1.3 Decoderen

1.2 Adaptive huffman

1.2.1 Algemene operaties

1.2.2 Encoderen

1.2.3 Decoderen

1.3 Adaptive huffman met sliding window

1.3.1 Algemene operaties

1.3.2 Encoderen

1.3.3 Decoderen

1.4 Two pass adaptive huffman

1.4.1 Algemene operaties

1.4.2 Encoderen

1.4.3 Decoderen

1.5 Bloksgewijze adaptive huffman

1.5.1 Algemene operaties

1.5.2 Encoderen

1.5.3 Decoderen

2. Datasets

3. Experimenten

4. Besluit

4.1 Statische huffman

4.2 Adaptive huffman

4.3 Adaptive huffman met sliding window

4.4 Two pass adaptive huffman

4.5 Bloksgewijze adaptive huffman