Käyttöohje

1 Ajaminen

1.1 Komentoriviargumentit

- Jar-tiedosto on `tl15/dist/`-hakemistossa, missä ohjelman voi ajaa komennolla `java -jar tl15.jar [argumentit]`.
- Argumentti-info tulostuu argumentilla `-h`: `java -jar tl15.jar -h`.
- Luettava tiedosto annetaan argumentilla `-i tiedosto.in` ja kirjoitettava argumentilla `-o tiedosto.out`. Jos nämä ovat sama tiedosto, niin ohjelma saattaa käyttäytyä arvaamattomasti.
- Oletuksena ohjelma pakkaa luettavan tiedoston kirjoitettavaan tiedostoon. Antamalla argumentti `-d` ohjelma sen sijaan purkaa luettavan tiedoston kirjoitettavaan tiedostoon.
- Pakkauksessa/purussa käytetty algoritmi voidaan vaihtaa argumentilla `-a algo`. Oletuksena on LZW ja Huffman saadaan käyttöön sanomalla `-a huffman`. Jos tiedosto on pakattu eri algoritmilla kuin sitä yritetään purkaa, niin ohjelma antaa virheilmoituksen. Ohjelma ei toistaiseksi osaa valita oikeata algoritmia pakatun tiedoston sisällön perusteella.
- Jos pakataan algoritmilla LZW, niin koodisanan maksimikoon voi valita argumentilla `-ls koko`, missä `koko` on kokonaisluku $9, \ldots, 31$. Tämä vaikuttaa eksponentiaalisesti ohjelman käyttämän muistin määrään,[1] joten käytännössä maksimikoon on (suurilla tiedostoilla) oltava reilusti alle 30. Oletuksena on 12. Ohjelma osaa tunnistaa LZW-pakatusta tiedostosta, mikä oli pakkauksessa käytetty koodisanan maksimikoko, joten tätä argumenttia ei purettaessa tarvita.

[1] Jos $n =$ `koko`, niin sanaston kooksi tulee 2^n sanaa. Pahimmassa tapauksessa i:nnen sanan pituus on i merkkiä, jolloin sanojen yhteenlaskettu pituus on $2^{n-1}(2^n + 1)$. Jokainen merkki voi viedä 2 tavua, joten sanojen viemä tila on n. $2^n(2^n + 1)$ tavua. Siis jos `koko` $= 12$ (oletus), niin sanasto vie pahimmassa tapauksessa n. 16MB ja jokainen lisäbitti suunnilleen nelinkertaistaa maksimikoon.

1.2 Esimerkkejä

- Pakkaa tiedosto `teksti.txt` tiedostoon `teksti.txt.hc` käyttäen Huffman-algoritmia:

```
java -jar tl15.jar -i teksti.txt -o teksti.txt.hc -a huffman
```

- Pura tiedosto `teksti.txt.hc` tiedostoon `teksti.txt` käyttäen Huffman-algoritmia:

```
java -jar tl15.jar -i teksti.txt.hc -o teksti.txt -a huffman -d
```

- Kuten äskeiset, mutta käytetään LZW-pakkausta (oletus) ja koodisanan maksimikokoa 12 (oletus):

```
java -jar tl15.jar -i teksti.txt -o teksti.txt.lc
java -jar tl15.jar -i teksti.txt.lc -o teksti.txt -d
```

- Kuten äskeiset, mutta käytetään LZW-pakkausta ja koodisanan maksimikokoa 16:

```
java -jar tl15.jar -i teksti.txt -o teksti.txt.lc -ls 16
java -jar tl15.jar -i teksti.txt.lc -o teksti.txt -d
```

2 Testien ajaminen

Yksikkötestit voi ajaa vaikka NetBeansista tavalliseen tapaan. Suorituskyky/vertailutestejä voi tehdä `tl15/`-hakemiston `runtest`-skriptillä. Se olettaa, että hakemistossa on `test.orig`-niminen tiedosto. Skripti pakkaa tämän Huffmanilla, LZW:llä muutamalla eri koodisanan koolla sekä bz2-ohjelmalla. Jokaisesta tulostetaan aika- ja kokostatistiikkaa, mutta muutakin ruudulle tulee, joten se on käsin parsittava.