

PackerX - Toteutusdokumentti

Timo Mäki

7. syyskuuta 2014

1 Toiminta

Ohjelman toiminta perustuu kahteen luokkaan, HuffmanCompressor ja HuffmanDecompressor. Compressor luo huffmanpuun annetusta tekstitiedostosta ja kirjoittaa sen levyille. Myös tekstin binääriesitys kirjoitetaan. Decompressor lukee sekä puun että binääriesityksen levyiltä. Tämän jälkeen se muuttaa binäärimuodossa olevan tekstini takaisin luettavaksi.

2 Aika vaativuus

Compressor toimii ajassa $O(N + M \log(M) + N \cdot M)$, missä N on tekstin pituus ja M on eri kirjainten määrä. Tekstin pituuteen verrannollisuus tulee siitä, että koko tekstitiedosto täytyy lukea. Koska jokainen kirjain asetetaan kekkoon omaksi lehdekseen ja kekkoon lisäys on logaritminen aikavaatimukseltaan, käyttää ohjelma siihen $\log(M)$ aikaa. Koska operaatio suoritetaan jokaiselle lehdelle, on kyseessä $M \log(M)$ operaatio. Kirjoitus vaiheessa käydään koko teksti läpi, ja muutetaan se binääri esitykseksi. Tällöin pituus on N ja binääri esitys on tallennettu BitSet:iin, jonka pituus riippuu puun eri kirjainten määrästä; enemmän oksia johtaa pidempään esitykseen.

Compressorissa on muitakin iteraatioita, mutta ne toimivat edellä mainituissa ajoissa, joten ne muuttavat vain kerrointa. Esimerkiksi serialisoinnissa menee aikaa, mutta voimme nähdä, että N kokoisen tekstin, jossa on M kirjainta, tallennuksessa menee $O(N \cdot M)$ aikaa.

Decompressor toimii ajassa $O(N \cdot M + N \log(M))$. Aika vaatimus $N \cdot M$ tulee binääriesityksen luku pituudesta, kun siinä on N pituinen teksti jossa on M kirjainta. $N \log(M)$ tulee taas kun jokainen kirjain täytyy muuttaa luettavaan muotoon, ja kirjain on haettava huffmanpuusta, joka toimii $\log(M)$ ajassa.

3 Modulaarisuus

Ohjelma on suunniteltu jokseenkin modulaariseksi. DataRepresentation ja Collections paketeista ei ole riippuvuuksia muihin ohjelman osiin. Compressors paketista on riippuvuuksia Collections sekä DataRepresentation paketteihin. PackerX paketista on riippuvuuksia Compressor ja DataRepresentation paketteihin.

Compressorit heittävät exceptioneja virhetilanteissa. Ne myös mahdollistavat annettavaksi PrintStream objektin konstruktorissaan, johon ne voivat kirjoittaa tietoa ohjelman suorituksesta. Tämä mahdollistaa kehityksen siten, että voidaan luoda myös graafinen käyttöliittymä ohjelmalle.