

PackerX - Toteutusdokumentti

Timo Mäki

24. elokuuta 2014

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 binäärimuodon takaisin luettavaksi.

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 kekoon omaksi lehdekseen ja kekoon 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.