

Määrittelydokumentti

Alexis Varjo

Kieli: c++

Opinto-ohjelma: TKT

Käytän projektissa C++:aa, koska haluan oppia kyseistä kieltä enemmän, niin pyrin tekemään kaiken koodauksen tällä hetkellä sillä kielellä. Siinä on myös valmiita tietorakenteita, hyvä mahdollisuus bittitason tiedon käsittelyyn, ja se on laitteistoa lähellä oleva kieli, joten sillä koodatut ohjelmat käyttävät yhden ytimen tehoa kätevästi.

Muita riittävästi hallitsemiani kieliä on esimerkiksi Python ja C, joilla tehtyjä töitä pystyn arvioimaan.

Toteutetut algoritmit: Huffmanin koodaus ja LZ77.

Ratkaistu ongelma on rajallisen tiedostotilan säästäminen.

Ohjelma saa syötteenä tiedoston, ja argumenttina ajanmittauksen (tai performance evaluation mode), ja käytetyn algoritmin.

Huffmanin koodauksessa aikavaativuuden analyysi ei ole merkittävässä osassa O-analyysinä, koska siinä algoritmissa n olisi tekstin sisältämien merkkien lukumäärä, joka pysyy suomen- ja englanninkielisissä teksteissä suhteellisen vakiona. Huffmanin koodaus vie tilaa noin 2 – 320 tavua, jos oletuksena on 8-bittiset aakkoset ja puu kootaan rekursiivisesti.

LZ77 toteutetaan liukuvan ikkunan avulla, ja sen aikavaativuus on $O(n \cdot w)$, jossa n on pakattavan tekstin koko, ja w on liukuvan ikkunan koko. Pakkaus on mahdollista optimoida $O(n)$ -vaativuuteen tilankäytön kustannuksella.

Harjoitustyön Ydin

Harjoitustyöni tärkein osa on pakkausalgoritmit. Tärkeintä on, että pakkaus toimii häviöttömästi ja sitä on helppo testata ja mitata eri syötteillä. Myös esimerkiksi muistinhallinta ja yleisesti ohjelman pitää toimia joka ikisellä kerralla. Ytimessä on LZ77/LZ78, ja Huffmanin koodaus. Toinen yhtä tärkeä osa ohjelmaa on ohjelman osa, joka purkaa pakkauksen.

Viitteet:

1: <https://stackoverflow.com/questions/759707/efficient-way-of-storing-huffman-tree>

2: https://en.wikipedia.org/wiki/Huffman_coding

3: https://en.wikipedia.org/wiki/LZ77_and_LZ78