

Aloitin tällä viikolla ohjelman suorituskky testauksen ja käytin testaukseen 3 eri tiedostoa(pieni tekstitiedosto, suuri tekstitiedosto, .html tiedosto. Nämä tiedostot löytyvät projektista kansioista “testfiles/suorituskky”. Alla ovat testauksen tulokset helposti luettavissa olevassa taulukossa.

Tiedoston nimi:	README.TXT	Bible.txt	Web_by_Design_HTML_Basics_Tutorial.html
Aluperäinen koko(tavua):	1329	5498995	70601
Koko huffmanille pakatessa(tavua)	1058	3068618	47714
Pakkauksen kesto huffmanin koodilla (ms)	19	21207	332
Pakkauksen purku huffmanin koodilla(ms)	19	21740	218
Koko LZWllä pakatessa (tavua)	2023	2156266	52648
Pakkauksen kesto LZWllä (ms)	18	471850	458
Pakkauksen purku LZWllä (ms)	16	475770	439

Kuten näkyy toteuttamani huffmanin koodaus on parempi kuin LZW kaikessa muussa kuin pakkauskoossa isossa tiedostossa. Lisäksi LZW suoritusaika sekä pakatessa että purettaessa näyttäisi kasvavan jostain syystä turhan pitkäksi, jotain ilmeisesti kusee algoritmista. Pienessä tekstitiedossa LZW algoritmille tuli myöskin niin paljon overheadia että “pakattu” tiedosto vei pakkauksen jälkeen enemmän tilaa kuin alkuperäinen. Tätä voisi ainakin vähentää kirjoittamalla pienessä tiedostossa koodit 12 bittisinä 24 bitin sijasta. Mutta tämä toisaalta lisäisi suoritusaikaa n:n verran...

Huffmanin koodin purkuaikaa voisi myös hieman parantaa jos onnistuisin kirjoittamaan pakattavan tiedoston alkuun itse huffmanin puun merkkien esintymisten sijasta. Sitten menisi puun rakentaminen nopeammin. Mutta ensin pitää saada LZW toimimaan paremmin ja mahdollisesti tehdä muita algoritmeja lisäksi.

Niin ja purkaminen jokaisessa tapauksessa onnistui niin että tiedosto säilyi samana.

Aion kahden viimeisen viikon aikana testata kaikkia algoritmeja ainakin muutamal muulla erityyppisellä tiedostolla.