Tiralabra 2015/periodi 3: Viikkoraportti 4

Aloitin tämän viikon tavoista poiketen testauksella. Koska Huffman-koodauksen ja -purun erikseen testailu oli melko epäluontevaa (testitiedostojen koodaaminen ”käsin” oli varsin työlästä, ja näin tehdyt tiedostot varsin yksinkertaisia), päädyin lisäämään molempien toimintaa yhdessä testaavan HuffmanTest-testiluokan. Testit pakkaavat/purkavat muutaman merkkijonon, minkä lisäksi automatisoin myös pakkauksen testaamisen aiemmin käytössä olleilla testitiedostoilla.

Toteutin tällä viikolla projektiin myös yhden uuden muunnoksen, ”move to frontin”(MTF). Muunnoksessa kaikki mahdolliset tavut (256kpl) ovat ensin listassa. Lähdetekstiä luetaan tavu kerrallaan, ja muunnetussa tiedostossa tavu korvataan indeksillään tuossa listassa. Kunkin tavun käsittelyn jälkeen, vastaava tavu siirtyy kaikkien tavujen listan alkuun (mistä muunnoksen nimi). Toisin sanoen tekstissä usein esiintyvät tavut pysyvät tavulistan alkupäässä, ja pieniä indeksejä vastaavat tavut saadaan muunnetussa tiedostossa todella yleiseksi, mikä hyödyttää muita menetelmiä (esim Huffman). Itsessään muunnos ei ”pakkaa”, eli muunnettu tiedosto on kooltaan sama kuin alkuperäinen.

Testailin MTF:n toimintaa Huffmanin kanssa, osoittautui ettei käytännön esimerkeissä muunnos tuonut merkittävää lisää, itseasiassa ainakin Alice's adventures in wonderlandin koko jopa hieman kasvoi. Laadin myös testitiedoston (”sampleFiles/aaa.txt”), jonka pakkausta muunnoksen teoriassa pitäisi tehostaa merkittävästi, ja näin näkyikin tapahtuvan. Yleisemmässä tapauksessa muunnoksesta lienee enemmän hyötyä, kun pakkaukseen lisätään muitakin muunnoksia (esim. Burrows-Wheeler (BWT) luo tiedostoon pitkiä saman merkin toistoja, jotka ovat tietenkin MTF:lle eduksi).

Testailin tällä viikolla myös ohjelman suorituskykyä. Testitiedostoillani pakkaus kestää noin puolisen sekuntia, ja purku reilun sekunnin. Pitkillä tiedostoilla suoritusaika kasvaa vääjäämättä, koska tiedoston pakkaus vie vähintään lineaarisen ajan. Tällä hetkellä HuffmanEncoding ja -Decoding -luokissa on tulosteita suoritusajasta, en aio jättää näitä lopulliseen versioon, mutta käytin niitä lähinnä erottelemaan mihin pakkauksen osaan aika lähinnä kuluu. Esim. alice.txt:lle sain seuraavia tuloksia:

Time to count bytes: 0.381392 sec
Time to build HuffmanTree: 0.014815000000000002 sec
Time to write encodings: 0.005738 sec
Time to write data: 0.7776500000000001 sec
Total runtime: 1.200016 sec

Purku:
Time to read codewords from file: 0.0059770000000000005 sec
Time to decompress when the codewords have been obtained: 1.7786300000000002 sec
Total runtime: 1.795347 sec

Kuten havaitaan, koko tiedoston dataa käsittelevät lineaarisen ajan operaatiot dominoivat suoritusaikaa. Aikavaativuusluokka näissä on paras mihin voidaan päästä, mutta optimoimalla lienee vielä mahdollista parantaa melko paljon.

Edellämainittujen lisäksi aika ei tällä viikolla riittänytkään muuhun kuin viime viikolla toteutettujen tietorakenteiden paranteluun ja vertaispalautteen laatimiseen. Tutustuin myös BWT:n toimintaan, mutta koodia en ehtinyt sen osalta vielä juurikaan toteuttamaan. Ensi viikolla toteuttanen move to frontin purkupuolen ja toivottavasti myös Burrows-Wheelerin.