Tiralabra 2013 periodi III aihemäärittely - tiedon pakkaus

Mika Viinamäki

16. tammikuuta 2013

1 Toteutetut algoritmit

Työssä toteutetaan ainakin yksi tiedonpakkausalgoritmi, joka on LZW. Mikäli aikaa jää, toteutetaan kenties muitakin ja mahdollisesti yhdistellään montaa eri algoritmia — yksi toteutettava algoritmi LZW:n lisäksi voisi olla Huffman-koodaus.

LZW-toteutus tulee käyttämään aputietorakenteena ainakin hajautustaulua, joka implementoidaan itse. Muut tarpeelliset tietorakenteet ovat vielä vähän avoinna, joskin hajautustaulu-implementaatio tulee todennäköisesti käyttämään myös jonkinlaista linkitettyä listaa.

2 Ratkaistava ongelma

LZW pystyy häviöttömästi pakkaamaan (ja purkamaan) tietoa. Kuten useat muutkin häviöttämät tiedonpakkausalgoritmit, LZW on erityisen tehokas mikäli pakattavassa datassa esiintyy toistuvia rakenteita. Esimerkki tälläisestä datasta on esimerkiksi suomenkielinen teksti.

Kaikista tiedonpakkausalgoritmeista toteutettavaksi valitsin nimenomaan LZW:n, koska se:

- on hyvin tunnettu ja siten siitä löytyy runsaasti materiaalia
- vaikutti muutamasta tunnetusta vaihtoehdosta mielenkiintoisimmalta toteuttaa
- on (kai) laajuudeltaan sopiva tähän harjoitustyöhön — ei siis toivottavasti liian suppea eikä ainakaan liian laaja

LZW-algoritmi itsessään ei ota kantaa siihen, millainen tietorakenne algoritmin perustana oleva sanakirja oikein on — hajautustaulu vaikuttaa kuitenkin luontevalta valinnalta.

Lisäksi alustavan LZW-kyhäelmän perusteella vaikuttaa siltä, että osa ratkaistavasta ongelmasta on bittien kanssa taistelu (mitä LZW vaatii) Javalla ja siihen liittyvät algoritmit.

3 Ohjelma ja syötteet

Ohjelman on tarkoitus vastaanottaa binäärimuotoista dataa `stdin`:stä ja pulauttaa pakattu/purettu data ulos `stdout`:sta. Tiedoston lukeminen tai sellaiseen tallennus ei sinällään ole kiinnostuksen kohteena — Linuxin (ja ymmärtääkseni myös Windowsin) komentotulkilla pystyy halutessaan helposti ohjaamaan tiedostosta dataa pakattavaksi tai purettavaksi ja myös ohjaamaan tiedostoon ohjelman tulosteen.

4 Suorituskyky

Itse LZW:n- aika tai tilavaativuudelle ei ole ainakaan toistaiseksi tarkkaa tavoitetta — tavoitteena on lähinnä saada aikaan ohjelma, joka pystyy pakkaamaan ja purkamaan suuriakin määriä dataa järkevässä ajassa.

Käytetyille aputietorakenteille on tavoitteena saada kullekin tietorakenteelle tyypillinen aika- ja tilavaativuus — esimerkiksi hajautustaulun tapauksessa aikavaativuudeksi lisäykselle tasoitetusti $\mathrm{O}(1)$.

5 Lähteet

`http://en.wikipedia.org/wiki/LZW`