Tiralabra 2013 periodi III aihemäärittely - tiedon pakkaus

Mika Viinamäki

15. tammikuuta 2013

1 Toteutetut algoritmit

Työssä toteutetaan ainakin yksi tiedonpakkausalgoritmi, joka on LZW. Mikäli aikaa jää, toteutetaan kenties muitakin — todennäköisenä kanditaattina on Huffman-koodaus.

LZW-toteutus tulee käyttämään aputietorakenteena ainakin hajautustaulua, joka implementoidaan itse. Muut tarpeelliset tietorakenteet ovat vielä vähän avoinna — hajautustaulu tosin kaipaa ainakin linkitetyn listan. ArrayList-tyylinen tietorakenne voi myös osoittautua hyödylliseksi.

2 Ratkaistava ongelma

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

Kaikista tiedonpakkausalgoritmeista nimenomaan toteutettavaksi valitsin LZW:n, koska se vaikutti laajalti dokumentoiduista ja tunnetuista vaihtoehdoista mielenkiintoiselta mutta toisaalta laajuudeltaan sopivalta tähän harjoitustyöhön.

LZW-algoritmi itsessään ei ota kantaa siihen, millainen tietorakenne algoritmin perustana oleva sanakirja oikein on — käytän tässä roolissa hajautustaulua, koska se vaikuttaa luontevalta valinnalta.

Lisäksi alustavan LZW-kyhäelmän perusteella tuntuu siltä, että osa ratkaistavasta ongelmasta on bittien kanssa taistelu (mitä LZW vaatii) Javalla. Näistä haasteista kuitenkin selvittäneen.

3 Ohjelma ja syötteet

Ohjelman on tarkoitus vastaanottaa binäärimuotoista dataa `stdin`:stä ja pulauttaa pakattu (tai purettu, parametreista riippuen) versio datasta 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 pakatun tai puretun datan tiedostoon.

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`