Huffman pakkaus- ja purkuohjelma

Määrittelydokumentti

Aineopintojen harjoitustyö: Tietorakenteet ja algoritmit (IV periodi)

Katri Roos

opiskelijanumero: 012729395

Helsingin yliopisto

14.3.2013

1. Ohjelman algoritmi ja sen toiminta

Teen pakkausohjelman käyttäen Huffmanin algoritmia. Algoritmin ideana on pakata tieto pienempään tilaan. Ensin luetaan dataa merkki kerrallaan ja lasketaan merkeille esiintymistiheys eli monta kertaa mikin merkki esiintyy datassa.

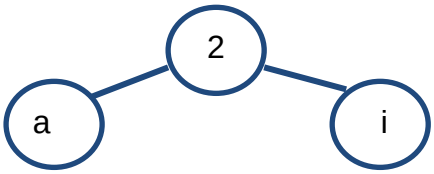
Esimerkki 1.1

”kissa”

a	1
i	1
k	1
s	2

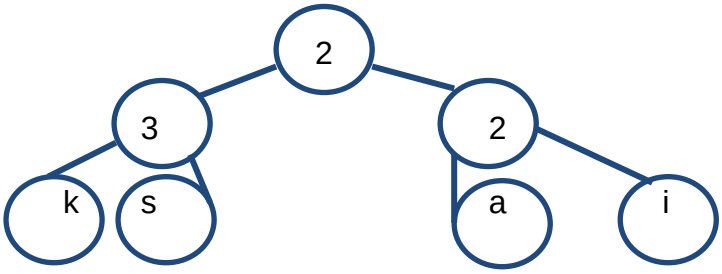
Esiintymistiheyksistä tehdään puun solmuja ja ne laitetaan nousevaan järjestykseen. Kahdesta pienimmästä solmusta tehdään binääripuu seuraavasti

Esimerkki 1.2



Merkit siis yhdistetään ja vanhemman arvoksi tulee yhdistettyjen merkkien tiheydet yhteen laskettuna. Tämä puu lisätään takaisin jonon, missä solmut ovat järjestyksessä. Tätä toistetaan, kunnes jonossa on vain yksi solmu, mikä muodostaa Huffmanin puun.

Esimerkki 1.3



Seuraavaksi jokaiselle merkille lasketaan uusi koodi puun avulla. Vasemmalle mentäessä otetaan nolla ja oikealle mentäessä 1 eli k:n koodi olisi 00. Mitä useammin merkki esiintyy, sitä lyhyempi koodi sille tulee eli data tiivistyy.

Esimerkki 1.4

a	10
i	11
k	00
s	01

Purkuohjelma toimii Huffmanin algoritmin periaatteen mukaisesti myös. Tiedostosta luetaan pakattu koodi ja sen pakkaukseen käytetyn puun tiedot. Pakattu koodi muutetaan takaisin merkeiksi puuta lukemalla juuresta lehtiin, kunnes koko pakattu koodi on käyty läpi.

2. Valitut tietorakenteet

Ohjelmassa käytetään binääripuita luomaan Huffmanin puu. Puun solmujen järjestämiseen käytetään prioriteettijonoa. Prioriteettijono on jono, mikä laittaa siihen lisätyt oliot, numerot yms. järjestykseen. Uusien koodien listaukseen ja esiintymistiheyksien laskuun käytetään tavallisia taulukoita.

3. Ohjelman syötteet

Pakkausohjelma saa syötteenä tiedoston nimen ja polun tekstinä. Kyseinen tiedosto haetaan, luetaan ja siitä muodostetaan pakattu koodi. Uusi tiiviimpi koodi tallennetaan toiseen tiedostoon, mille annetaan sama nimi, mutta uusi pääte. Ohjelma voisi saada syötteenä myös esimerkiksi uuden kohdeosoitteen, mihin pakattu tiedosto tallennetaan.

Purkuohjelma saa syötteen pakatun tiedoston nimen ja polun tekstinä. Tiedostosta luetaan puun tiedot ja pakattu koodi. Koodi muutetaan takaisin merkeiksi ja ohjelma tallentaa merkit samannimisenä kuin alkuperäinen tiedosto. Tässäkin voisi antaa uuden kohdeosoitteen syötteenä mahdollisesti.

4. Tavoitteena olevat aika- ja tilavaativuudet

Huffmanin puun generoimisen aikavaativuus on O(n log n). Pakattavia merkkejä on n kappaletta ja niistä muodostetaan binääripuita kunnes jäljellä on yksi.

Pakkaaminen ja purkaminen vie aikaa O(m * n), koska käydään läpi jokainen merkki m ja se käydään läpi n kertaa, mikä on uuden koodin pituus.

5. Lähteet

http://www.cprogramming.com/tutorial/computersciencetheory/huffman.html

http://www.compressconsult.com/huffman/#conventions

http://fi.wikipedia.org/wiki/Huffmanin_koodaus