Datan pakkaus - määrittelydokumentti

Toteutan tietorakenteiden ja algoritmien harjoitustyönä dataa pakkaavan java-ohjelman. Alustavasti ohjelma tulee sisältämään tekstimuotoisen datan Huffman-koodauksen, josta laajennan sitten muiden tiedostotyyppien pakkaukseen ja monimutkaisempiin, esimerkiksi Huffmania välivaiheena käyttäviin menetelmiin käytettävissä olevan ajan mukaan. Ohjelman käyttö tulee tapahtumaan komentoriviltä. Tarkka käskyjen muoto täytyy vielä suunnitella, mutta ohjelma voisi toimia esimerkiksi seuraavasti:

```
java Main c pakattavaTiedosto.txt mihinPakataan
java Main d purettavaTiedosto mihinPuretaan.txt
```

Huffman-koodauksen toteutus vaatii ensinnäkin Huffman-puun rakentamisen. Huffman-puu on erityinen tietorakenne, jossa jokaisella sisäsolmulla on kaksi lasta. Huffman-puun rakentaminen vaatii edelleen uusia tietorakenteita. Puuta rakennetaan siten, että jokainen lehdeksi jäävä solmu (jotka vastaavat pakattavassa tekstissä esiintyviä symboleita) on ensin oma alipuunsa, ja kaksi harvinaisinta alipuuta (ne, joiden sisältämien merkkien yhteenlaskettu esiintymistiheys on pienin) yhdistetään aina uudeksi alipuuksi. Alipuita tulee helpon rakentamisen ja nopeiden lisäysten ja poistojen vuoksi pitää järjestetyssä listassa. Toteutettavia tietorakenteita ovat siis ainakin Huffman-puu ja solmujen tallentamiseen käytettävä lista.

Kun datan pakkaus alkaa, täytyy ensin laskea merkkien esiintymismäärät tekstitiedostossa. Tämä voidaan toteuttaa ajassa O(n). Tämän jälkeen suoritetaan Huffman-puun rakentaminen. Huffman-puuhun tulee m lehteä, missä m on tekstitiedoston aakkoston koko (esim. 128 jos oletetaan pelkkien ASCII-merkkien käyttö). Jokainen alipuiden yhdistäminen vähentää yhdistettävien alipuiden määrää yhdellä, ja puun rakennus on valmis kun puita on enää yksi. Puun rakentamisessa suoritetaan alipuiden yhdistämistä siis m-1 kertaa. Aikavaativuus ei kuitenkaan ole O(m), sillä jokaisen yhdistämisen jälkeen luotu uusi alipuu täytyy asettaa alipuiden listassa oikeaan kohtaan, mikä on myös O(m) operaatio. Kokonaisaikavaativuus onkin siis $O(m^2)$.

Kun Huffman-puu on rakennettu, aakkoston merkkien koodisanat voidaan lukea puusta käymällä jokainen solmu läpi kerran. Huffman-puun solmujen määrä on luokkaa O(m), joten myös tämän osion aikavaativuus on O(m). Kun koodisanat ovat valmiit, pakattava teksti saadaan koodattua ajassa O(l*n), missä l on pisimmän merkkikohtaisen koodisanan pituus.