Tehtävä 2: Tiedon tiivistäminen

Tutustu tarkoin seuraavaan tekstiin ja vastaa sitä hyväksi käyttäen tehtävän loppuosassa esitettyihin neljään kysymykseen.

Merkki on kirjoituksen alkeisosanen. Merkkejä ovat muun muassa kirjaimet, numerot, välimerkit ja erikoismerkit, kuten esimerkiksi pykälän merkki (§). Tietokoneella tekstiä esitettäessä tieto siitä, mikä merkki on kyseessä, joudutaan koodaamaan jollakin tapaa. Tyypillisesti tietokoneissa merkit esitetään kiinteänpituisina bittijonoina. Kahdeksan bittiä on tavanomainen yhden merkin koodin pituus.

Kirjoitetuissa teksteissä erilaisten merkkien esiintymistiheys vaihtelee. Tätä voidaan käyttää hyväksi tiedon tiivistyksessä. *Huffman-koodauksessa* yleisimpiä merkkejä vastaa lyhyempi bittijono, kun taas harvinaisempia merkkejä esitetään pitemmillä jonoilla. Tällä menettelytavalla on se etu verrattuna kiinteänpituisiin bittijonoihin, että tieto saadaan esitettyä kokonaisuudessaan pienemmällä bittimäärällä. Koska tiivistetty tieto vie vähemmän tilaa, saavutetaan säästöä esimerkiksi tiedon siirrossa ja talletuksessa.

Ennen varsinaista tekstin koodausta pitää merkeille johtaa sopivat bittijonoesitykset. Huffman-menetelmässä nämä muodostetaan käyttäen puumaisia kaavioita, joita kutsutaan *Huffman-puiksi*. Puun haarautumiskohtia kutsutaan *haarasolmuiksi* ja alimmaisten haarojen päässä olevia (kuvissa neliönmuotoisia) solmuja kutsutaan *lehtisolmuiksi*. Puu piirretään yleensä ylösalaisin, jolloin lehdet ovat alhaalla. Lehtisolmuja lukuunottamatta kaikilla solmuilla on tarkalleen kaksi *lapsisolmua*.

Huffman-puun muodostaminen aloitetaan laatimalla *lukumäärätaulukko* (kts. taulukko 1 sivulla 2). Huffman-puu muodostetaan lukumäärätaulukon perusteella käyttäen seuraavaa menetelmää:

1. Muodosta jokaista merkkiä kohti neliönmuotoinen lehtisolmu (kts. vaihe 1 kuvassa 1 sivulla 3), johon on liitetty itse merkki ja kyseisen merkin esiintymislukumäärä. Solmut on pidettävä esiintymislukumäärien mukaisesti kasvavassa järjestyksessä. Jos useammalla solmulla on sama esiintymislukumäärä, niin tällaisten solmujen keskinäisellä järjestyksellä ei ole merkitystä.

2. Toista kohtia (a) ja (b), kunnes on muodostunut kaikki solmut sisältävä yhtenäinen puu:

   (a) Ota käsittelyyn kaksi sellaista solmua, joiden esiintymislukumäärät ovat pienimmät ja jotka eivät vielä ole minkään toisen solmun lapsia.

(b) Luo uusi haarasolmu (ympyrä kuvassa 1), jonka lapsisolmuiksi tulevat (a)-kohdassa valitut solmut. Aseta tämän solmun esiintymislukumääräksi lapsisolmujen esiintymislukumäärien summa.

Huomaa, että muodostuneen puun lehtisolmuina ovat nyt tekstissä esiintyviä merkkejä vastaavat neliöt ja ylimpänä on niin kutsuttu *juurisolmu*, jonka arvona on kaikkien merkkien yhteislukumäärä. Puun jokainen vasen haara vastaa bittiä 0 ja jokainen oikea haara bittiä 1. Merkin koodi saadaan kulkemalla puu ylimpänä olevasta juurisolmusta alas koodattavaa merkkiä vastaavaan neliönmuotoiseen lehtisolmuun ja poimimalla bitit kuljetun reitin varrelta.

Huffman-koodilla on lisäksi sellainen ominaisuus, että minkään merkin koodin bittijono ei esiinny sellaisenaan minkään toisen koodin alkuosana. Tämä merkitsee sitä, että koodijono on aina yksiselitteinen.

Tarkastellaan nyt kuvassa 1 esitettyä yksityiskohtaista Huffman-puun rakentumista. Tarkastelun kohteena oleva teksti sisältää vain merkkejä `a`–`e` sekä välilyöntejä. Merkkien esiintymislukumäärät tekstissä on esitetty taulukossa 1. Viimeinen sarake tarkoittaa välilyöntien määrää tekstissä.

merkki	`a`	`b`	`c`	`d`	`e`	
lukumäärä	32	9	21	15	4	18

Taulukko 1: Merkit ja niiden esiintymislukumäärät

Vaiheessa 1 jokaista merkkiä kohti luodaan neliönmuotoinen lehtisolmu, joka sisältää merkin ja merkin esiintymislukumäärän. Lehtisolmut järjestetään esiintymislukumäärien mukaisesti kasvavaan järjestykseen. Merkit `e` ja `b` sisältävillä solmuilla on nyt pienimmät esiintymislukumäärät 4 ja 9. Menetelmän mukaisesti muodostetaan uusi haarasolmu, jossa esiintymislukumääränä on summa $4 + 9 = 13$ ja tämän solmun lapsina ovat merkit `e` ja `b` sisältävät solmut.

Vaiheessa 2 ne solmut, jotka eivät vielä ole minkään toisen solmun lapsia, on järjestetty uudelleen kasvavaan järjestykseen. Nyt pienimmät esiintymislukumäärät ovat 13 ja 15. Luodaan uusi haarasolmu, jossa on arvo $13 + 15 = 28$ ja lapsina esiintymislukumääriä 13 ja 15 vastaavat solmut.

Vaiheessa 3 pienimmät esiintymislukumäärät ovat välilyöntiä ja merkkiä `c` vastaavilla lehtisolmuilla. Muodostetaan uusi haarasolmu, jossa on esiintymislukumääränä summa $18 + 21 = 39$ ja lapsina välilyöntiä ja merkkiä `c` vastaavat solmut.

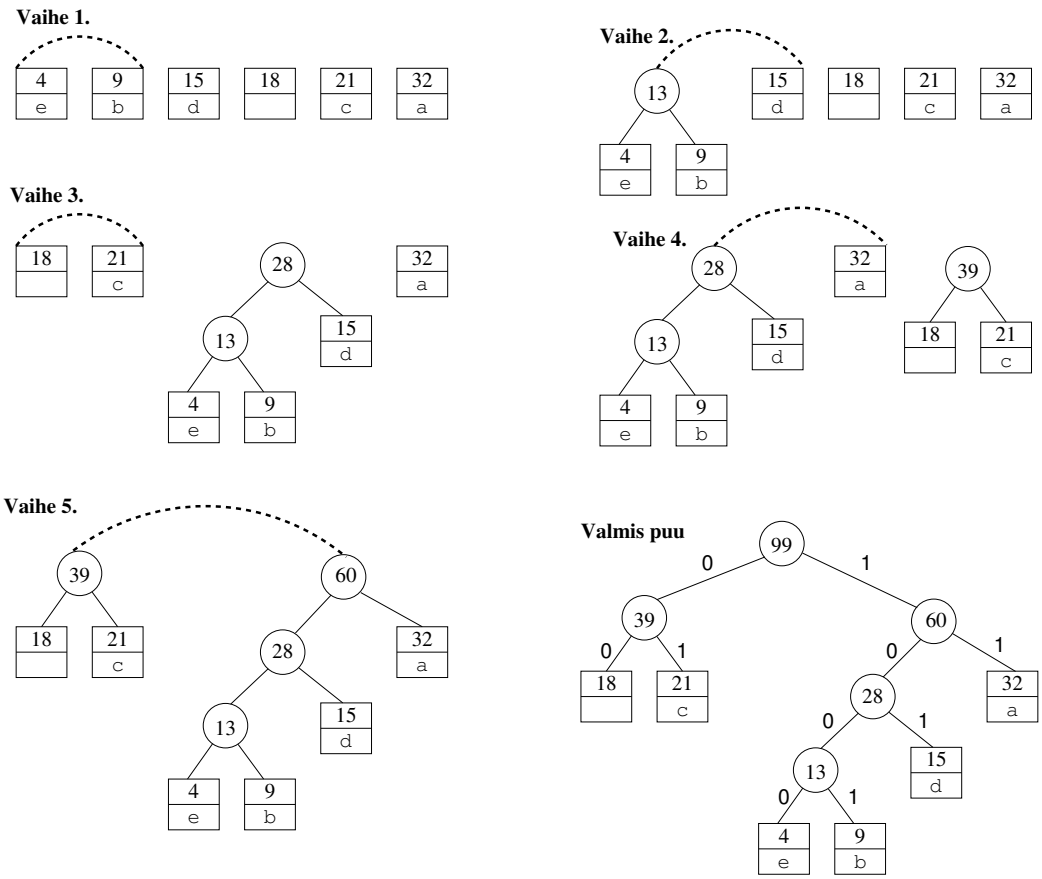
Vaiheessa 4 pienimmät esiintymislukumäärät ovat 28 ja 32. Luodaan uusi haarasolmu, johon asetaan esiintymislukumääräksi summa $28 + 32 = 60$ ja asetetaan tämän solmun lapsiksi asianmukaiset solmut.

Vaiheessa 5 muodostuu lopullinen puu. Puun juureksi tulee solmu jossa on esiintymislukumäärä 99. Se on tekstissä esiintyvien merkkien kokonaislukumäärä.

Puun haaroihin lisätään vielä bitit 0 ja 1 siten, että jokainen vasen haara vastaa bittiä 0 ja jokainen oikea haara bittiä 1.

Esimerkiksi merkki `a` koodataan jonoksi 11 ja välilyöntiä vastaa jono 00. Merkki `b` on koodattu jonoksi 1001, joka on pidempi kuin useammin esiintyvän merkin `d` koodi 101.

Merkkijonoja koodattaessa merkit muutetaan yksi kerrallaan vastaavaksi bittijonoksi. Esimerkiksi merkkijono `abc` koodataan bittijonoksi 11 1001 01. Vastaavasti bittijono 1001 11 1001 1000 vastaa sanaa `babe`.



Kuva 1: Huffman-puun muodostaminen taulukon 1 perusteella

Kysymykset

Kysymys 1. Koodaa 7-merkkinen teksti `abba cd` käyttäen kuvassa 1 esitettyä valmista Huffman-puuta. (maksimipistemäärä 4)

Kysymys 2. Mitä tekstiä vastaa kuvassa 1 esitettyä valmista Huffman-puuta käyttäen koodattu bittijono

1100100110001010011101101100010 1 ?

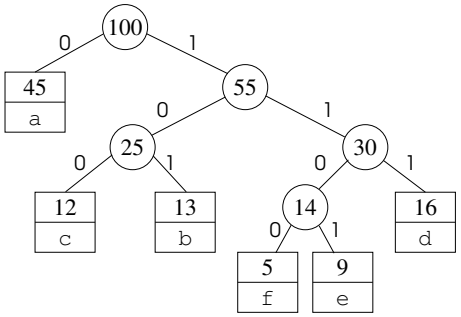
(maksimipistemäärä 4)

Kysymys 3. Eräs 100 000 merkkiä käsittävä tiedosto sisältää vain merkkejä `a`–`f`. Koska merkkejä on ainoastaan kuusi erilaista, ne on mahdollista koodata käyttäen kiinteänpituista 3-bittistä esitystä. Merkkien esiintymislukumäärät *tuhansissa* ja merkkejä vastaavat kiinteänpituiset koodibittijonot on esitetty taulukossa 2.

merkki	`a`	`b`	`c`	`d`	`e`	`f`
lukumäärä (tuhatta)	45	13	12	16	9	5
bittijono	000	001	010	011	100	101

Taulukko 2: Merkit, niiden esiintymislukumäärät ja 3-bittiset koodit

Kuvassa 2 on esitetty taulukon 2 merkkimääriä vastaava Huffman-puu.



Kuva 2: Taulukon 2 perusteella muodostettu Huffman-puu

Montako bittiä mainitun 100 000:n merkin tiedoston talletuksessa säästetään, kun käytetään kiinteänpituisten 3-bittisten koodien sijasta Huffman-koodausta? Esitä, miten säästö lasketaan ja laskun tulos. Huffman-puun vaatimaa tilaa ei tarvitse huomioida. (maksimipistemäärä 7)

Kysymys 4. Muodosta 20-merkkisen tekstin `a bearable barbarian` perusteella Huffman-puu ja koodaa kyseinen teksti käyttäen Huffman-koodausta. (maksimipistemäärä 10)