Tietorakenteet, laskuharjoitus 1, 19.-22.1

Huom: laskarit alkavat jo ensimmäisellä luentoviikolla

1. Halutaan todistaa, että oletuksesta A seuraa väite B eli propositiologiikan merkinnöin $A \to B$. Todistamiseen on useita erilaisia tekniikoita. Se, mikä kulloinkin sopii parhaiten, riippuu tilanteesta.

   - *Deduktiivisessa todistuksessa*, jota kutsutaan myös suoraksi todistukseksi, oletetaan A ja näytetään, että tästä seuraa B.

     Esim. Olkoon $x \in \mathbb{N}$ pariton. Väite: x^2 on pariton.
     Tod.: Jos $x \in \mathbb{N}$ on pariton (oletus), niin $x = 2k+1$ jollain $k \in \mathbb{N}$ (parittomuuden määritelmä). Silloin $x^2 = 4k^2 + 4k + 1 = 2(2k^2 + 2k) + 1,$ eli x^2 on pariton. $\square$
   - *Kontrapositiivisessa todistuksessa* näytetään että oletuksesta ei B seuraa ei A, eli että $\neg B \to \neg A$.

     Esim. Väite: Jos $x > 0$ on irrationaaliluku, niin $\sqrt{x}$ on irrationaaliluku.
     Tod.: Osoitetaan, että jos $\sqrt{x}$ on rationaaliluku, niin x on rationaaliluku. Siis $\sqrt{x} = p/q$, joillakin $p, q \in \mathbb{Z}, q \neq 0$, joten $x = \frac{p^2}{q^2}$ ja $p^2, q^2 \in \mathbb{Z}, q^2 \neq 0$. $\square$

   - *Vastaväitetodistuksessa* näytetään että oletuksesta $A \wedge \neg B$ seuraa ristiriita. Tekniikkaa kutsutaan myös epäsuoraksi todistamiseksi.

     Esim. Väite: Jos $a + b = 2$, niin $a \geq 1$ tai $b \geq 1$.
     Tod.: Oletetaan, että $a + b = 2$ ja tehdään vastaväite $a < 1$ ja $b < 1$. Tällöin $2 = a + b < 1 + 1 = 2$, ristiriita. $\square$

   Perustele sekä matemaattisesti että maalaisjärjen tasolla miksi kontrapositiivinen ja vastaväitetodistus ovat matemaattisesti hyväksyttävä tapa todistaa $A \to B$.

   Perustele myös miksi riittää näyttää että $A \to B$ ja $B \to A$ jos halutaan todistaa että väittämät A ja B ovat yhtäpitävät eli $A \leftrightarrow B$.

   Kurssin Matematiikka tutuksi viikon 6 materiaali voi auttaa asiaan
   http://wiki.helsinki.fi/display/mathstatKurssit/MaTukurssisivu

   Ylläolevista erityisesti **vastaväitetodistus on kurssin aikana aktiivisessa käytössä**, eli vaikka tehtävänanto on pitkä, lue se huolellisesti ja **yritä sisäistää todistustekniikat mahdollisimman hyvin**.

2. Vastaväitetodistuksen lisäksi **induktiotodistus** on Tietorakenteissa avainasemassa.

   Tirassa hyödynnetään usein erilaisia summakaavoja. Esim. kaava $\sum_{i=0}^{n} i = 1 + 2 + 3 + \ldots + n = \frac{n(n+1)}{2}$ on kovassa käytössä. Summakaava on yleensä helpointa todistaa induktiolla.

   Todista, että $\sum_{i=0}^{n} 2^i = 2^{n+1} - 1$

3. Tietojenkäsittelijä voi ajatella logaritmia usein seuraavasti: a-kantainen logaritmi $\log_a n$ kertoo, kuinka monta kertaa luku n pitää jakaa a:lla, ennen kuin päästään lukuun 1. Esimerkiksi $\log_2 8$ on 3, koska $8/2/2/2$ on 1. Jos lukuun 1 ei päästä tasajaoilla, logaritmin lopussa on desimaaliosa, joka kuvaa viimeistä vaillinaista jakoa. Esimerkiksi $\log_2 9$ on noin 3,17, koska $9/2/2/2$ on 1,125, eli kolmen jaon jälkeen on vielä hieman matkaa lukuun 1, mutta neljäs täysi jako veisi jo selvästi sen alapuolelle.

   Laske seuraavat logaritmit ilman laskinta:

   (i) $\log_2 4$

   (ii) $\log_2 32$

   (iii) $\log_3 3$

   (iv) $\log_3 81$

   (v) $\log_7 1$

   (vi) $\log_7 49$

   (vii) $\log_{10} 1000$

   (viii) $\log_{10} 10000$

4. Jatkoa edelliseen tehtävään

   a. Perustele seuraava laskusääntö peräkkäisten jakamisten avulla:
   $\log_a(x * y) = \log_a x + \log_a y$

   b. Perustele seuraava laskusääntö edellisen laskusäännön avulla:
   $\log_a x^y = y * \log_a x$

5. Fibonaccin lukujonon n:s luku voidaan määritellä seuraavasti

$$F(n) = \begin{cases} 1 & \text{jos } n = 1 \text{ tai } n = 2 \\ F(n-1) + F(n-2) & \text{jos } n > 2 \end{cases}$$

   Laske kynällä ja paperilla 10 ensimmäistä Fibonaccin lukua.

   Tee rekursiivinen (staattinen) metodi `long fib1(n)`, joka palauttaa Fibonaccin lukujonon n:nnen luvun. Rekursiivinen metodi on siis sellainen joka kutsuu itse itseään. Käytetään tässä tehtävässä Javaa.

   Tyypin `int` sijasta kannattaa käyttää tyyppiä `long` sillä lukujono kasvaa hyvin nopeasti.

   Fibonaccin lukujonon n:s luku on helppo selvittää myös ilman rekursion käyttöä

   Tee metodi `long fib2(n)`, joka palauttaa Fibonaccin lukujonon n:nnen luvun. Metodi käyttää laskennan apuna n:n mittaista taulukkoa.

   Edes taulukkoa ei välttämättä tarvita. Tee vielä taulukoton ilman rekursiota toimiva versio `long fib3(n)`.

   Mikä ohjelmista on nopein? Eniten muistia vievä? Entä selkein?

Huom: älä testaa kurssilla tekemiäsi ohjelmia koneissa melkki, melkinpaasi, melkinkari.

6. Olkoon x desimaaliluku ja $n \geq 0$ kokonaisluku. Luku x^n voidaan laskea seuraavan palautuskaavan mukaisella rekursiivisella algoritmilla:

$$x^n = \begin{cases} 1 & \text{jos } n = 0 \\ x x^{n-1} & \text{jos } n \text{ pariton} \\ x^{n/2} x^{n/2} & \text{jos } n \text{ parillinen} \end{cases}$$

Näytä miten kaavaa käyttäen lasketaan 3^{11}.

Laadi Javalla staattinen metodi `double potenssi(double x, int n)` joka toteuttaa algoritmin.

Kuten tehtävistä 4 ja 5 käy ilmi, rekursiivinen ratkaisu ei välttämättä ole kovin nopea. Miten on tämän tehtävän ratkaisun laita?