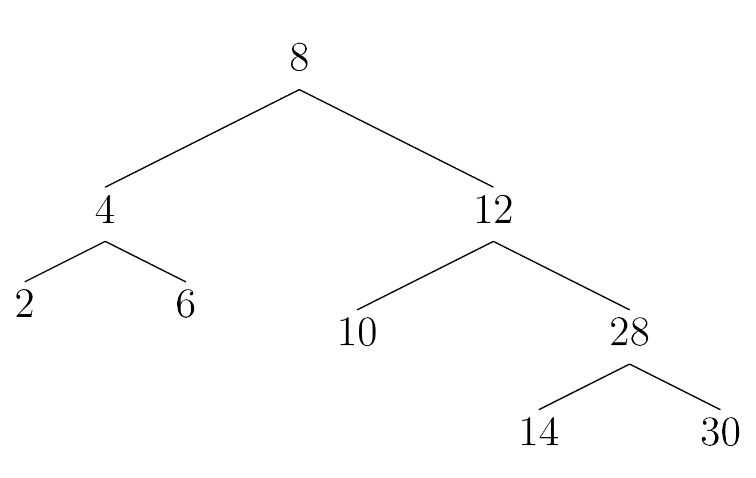
Tehtävänkuvaus

Työn tavoitteena on toteuttaa tietokoneohjelma, joka järjestää annetun aineiston tasapainoitettuun binääriseen hakupuuhun. Tämän jälkeen puusta haetaan aineistoa. Edelleen puuhun voidaan lisätä uutta aineistoa tai hakea aineistoa. Lisäykset ja haut tapahtuvat aineiston avaimen perusteella.

Kuvan 1. puussa avaimina ovat olleet **2, 4, 6, 8, 10, 12, 14, 30, 28**. Jokaisen avaimen on määrittänyt aineisto, mutta nyt vain avainten arvot on esitetty puussa. Tämä riittää myös harjoitustyössä, eli käsitellään vain avaimia.



Kuva 1. Tasapainotettu binäärinen hakupuu.

**Ohjeita**

Harjoitustyössä tehtävänä on tehdä ohjelma, joka

* lukee tiedostosta avaimet
* lisää luetut avaimet puuhun
* tulostaa syntyneen puurakenteen
* hakee avaimen mukaisen tiedon puusta (hakee avaimen)
* lisää yksittäisiä avaimia puuhun

Tietorakenteena on tasapainoitettu binäärinen hakupuu, jonka toteutus voi tapahtua taulukolla tai linkitetyllä rakenteella. **Ohjelman tulee antaa välitulosteita siten, että käyttäjä voi nähdä puun rakentumisen vaiheet**. Samoin yksittäiset lisäykset tulee havainnollistaa. Haku ilmoittaa avaimen olemassaolon puussa: joko avain löytyy tai sitten ei. Löytyykö puusta avaimet **6**, **1**, **10** ja **16**?

Käytä testiaineistona edellisen esimerkin avainjonoa annetussa järjestyksessä. Jatka sitten lisäämällä yksittäin avaimet **26**, **24**, **22**, **20**, **18**, **16** tässä annetussa järjestyksessä. Löytyykö avaimet **10**? ja **26**? Entäpä avain **32**?

Laadi oma avain-aineistosi, jolla demonstroit ohjelmasi toimintaa siten, että puun rakentamisen kaikki eri toiminnot tulevat esitellyiksi.

Puun tulostuksen tulee olla sellainen, että puun rakenne näkyy (avaimet tulostettuna puumaiseen muodostelmaan - grafiikkaa tms. ei vaadita). Vinkki: puun tulostus sivuttain (juurisolmu vasemmalla, oikea lapsi ylempänä, vasen alempana) on huomattavasti helpompi toteuttaa kuin pystyyn.

Pohdi seuraavia asioita:

* Miten voitaisiin puusta poistaa avain **4**? (poistoa ei tarvitse toteuttaa)
* Mikä on haun kompleksisuus? Toteutuuko esittämäsi kompleksisuus toteuttamassasi ohjelmassa? Testaa ohjelmaasi, kun avaimia on 10,100,1000,10000.

**Palautus**

**Harjoitustyö palautetaan henkilökohtaisessa tapaamisessa opettajan huoneessa 2425**. Tapaamisessa käydään läpi harjoitustyön toteutus ja ohjelman toiminta, joka opiskelijan tulee demonstroida. Ympäristönä on Linux, jossa on gcc-kääntäjä. Käännöksessä ei tietenkään saa tulla virheitä, mutta ei myöskään varoituksia kääntäjäoptioilla -Wall -ansi. **Keskeneräisiä töitä ei käsitellä tapaamisessa**.

Varaa palautusaika alla olevasta linkistä:

<https://moodle.lut.fi/mod/scheduler/view.php?id=226486>

**Itseopiskeluvaihtoehdon** suorittaja varaa samalta listalta 40 minuutin ajan ehtiäkseen esitellä sekä harjoitustyön että tehtävät samalla kertaa.

Materiaali hakupuista ja tasapainotetusta binäärihakupuusta luennolla 7.

Luento 8 harjoitukset ohjelman perustana

Neuvoja harjoitus 8 tehtävän annosta.

**RR- ja RL-kierrot tasapainotetussa binääripuussa** (4 p.)

Luentomateriaaleissa on esitetty C-funktio,  joka tekee binääripuun tasapainottavat LL- ja LR-kierrot. Täydennä oheinen ohjelma lisäämällä siihen funktio oikea\_kierto(puuosoitin \*emo, int \*etp), missä emo on osoitin puun solmuun (puuosoitin) ja etp on osoitin tasapainottomuuden ilmaisevaan int-tyyppiseen muuttujaan:

* Ohjelmoi RR- ja RL-kierrot, jotka ovat symmetrisiä LL- ja LR-kierroille.
* Valmistaudu esittelemään ohjelmasi koko toiminta ja ajamaan se annetulla syötteellä. Anna oma testisyöte.
* Ohjelman on esitettävä puun rakentuminen solmu solmulta siten, että kuhunkin kiertoon liittyvät toimenpiteet näkyvät. Tulosta siis puurakenne kussakin vaiheessa esittäen myös tasapainoilmaisimet.