

Ohjelman yleisrakenne

Jokainen keko on oma luokkansa, ja sen lisäksi keot toteuttavat erilaisia rajapintoja. Jokainen keko toteuttaa rajapinnan Keko, jota käytetään kekojärjestämisessä. Binäärikeko ja darykeko toteuttavat rajapinnan SolmutonKeko, ja Binomikeko sekä Fibonaccikeko toteuttavat rajapinnan SolmullinenKeko. Näiden rajapintojen kautta toimii Dijkstra.

Binäärikeko on, kuten tirassakin, tavallinen int-tyyppinen lista, ja pointteri heapSize, joka kertoo keon koon. Dijkstraa varten keossa on myös toinen int-tyyppinen lista, node, josta löytyvät verkon solmut. Rajapinnan SolmullinenKeko operaatiot päivittävät myös tätä listaa. Rajapinnan Keko metodit eivät sitä käytä. D-ary -keko on rakenteeltaan samanlainen kuin binäärikekokin.

Loput keot muodostuvat Solmuista. Jokaisella Solmulla on pointteri edelliseen ja seuraavaan solmuun, kuten myös lapsen ja vanhempaan (tosin yleensä ainakin yksi näistä on null). Binomikeko ja Fibonaccikeko ovat siis linkitettyjä listoja. Molemmilla on Solmu keko, pointteri ensimmäiseen alkioon. Fibonaccikeossa on lisäksi suoraan pointteri minimiin. Luokassa Solmu on ominaisuus marked, jota Binomikeko ei käytä, se on osa Fibonaccikekoa ja sen decreaseKey-metodia jossa sitä käytetään pitämään kekoa matalana. Luokassa solmu on myös ominaisuus node, jota käytetään Dijkstran algoritmin kanssa. Seuraavana havainnekuva siitä, miten solmut on linkitetty toisiinsa Binomikeossa (ja Fibonaccikeossa). Harmaat nuolet osoittavat null:iin.

