

Syvyys- ja leveyshaku toiminta ja käyttö, koodin dokumentaatio

Hanna Leppänen

Syvyyshaku

Syvyyshaku on graafialgoritmi, jolla käydään läpi graafin solmut edeten ensin niin pitkälle kuin mahdollista yhtä graafin haaraa. Kun viimeinen lapsisolmu on käyty läpi, palataan takaisin graafia, kunnes saavutetaan seuraava vierailematon haara, joka käydään taas loppuun ennen seuraavaan siirtymistä. Syvyyshaku sopii hyvin suunnatussa graafissa syklien havaitsemiseen ja labyrinttien reitin selvittämiseen.

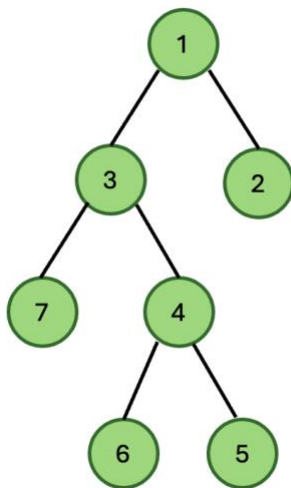
Leveyshaku

Leveyshaku on graafialgoritmi, jossa tietyllä syvyydellä juurisolmusta olevat solmut käsitellään ennen syvemmälle siirtymistä. Leveyshaussa käydään myös läpi kaikki solmut, kuten syvyshaussa. Leveyshaussa käytetään usein apurakenteena jonoa, jotta havaittuja mutta vierailemattomia lapsisolmuja voidaan seurata. Leveyshaku on hyvä algoritmi kahden solmun välisen lyhyimmän reitin etsimiseen.

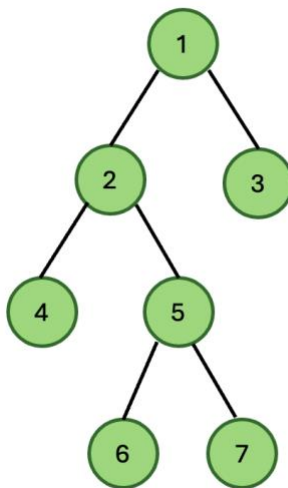
Esimerkkejä:

Puugraafissa molemmat algoritmit alkavat juurisolmusta ylhäältä. Solmut on numeroitu siihen järjestykseen, jossa niissä vieraillaan.

Syvyyshaku:

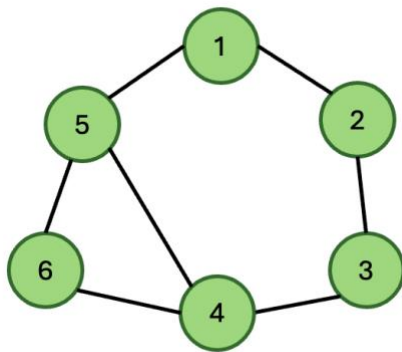


Leveyshaku:

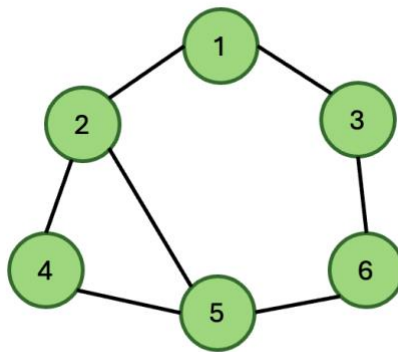


Suunnatussa graafissa haku aloitetaan valinnaisesta solmusta. Suunnatussa ja puugraafissa voidaan hyödyntää solmujen yhteyksien painoja suunnan valinnassa.

Syvyyshaku:



Leveyshaku:



Koodin dokumentaatio:

Ensin koodissa alustetaan pygame -kirjaston käyttöön tarvittavia muuttujia kuten ikkunan asetukset, värit ja solmujen sijainnit ikkunassa.

`draw.graph()` funktio piirtää graafin pygame ikkunaan. Graafin solmut iteroidaan läpi ja piirretään yksitellen pygame-kirjaston funktiolla `pygame.draw.circle()`. Solmujen väliset viivat piirretään `pygame.draw.line()` funtiolla.

`dfs()` funktio suorittaa syvyysshaun. Funktiossa käytetään pinoa (last-in-first-out) algoritmin suorittamiseen alussa alustetulle graafille.

`bfs()` funktio suorittaa leveyshaun. Tietorakenteena funktiossa käytetään jonoa (first-in-first-out).

Molemmissa algoritmin suorittavissa funktioissa ensin testataan, onko nykyisessä solmussa (`current_node`) jo käyty vai ei. Jos solmussa ei ole käyty se tarkistetaan ja siirrytään algoritmin mukaisesti seuraavaan solmuun.

`draw_visited()` funktiolla värjätään vieraillut solmut siniseksi.

`main()` funktiossa kutsutaan muita funktioita. Lisäksi toiminnallisuus, jossa näppäintä `s` ja `l` painamalla eri hakualgoritmit käynnistyvät on toteutettu tässä pygamen `keydown` komennolla.

Kehitysideat:

- minkä tahansa graaafin toiminnan mallintaminen
- käyttäjän syöttämä graafi
- graafin muokkaaminen suoraan pygame ikkunassa