

Käyttöohje

Ohjelman voi halutessaan kääntää jar-tiedostoksi, mutta ohjelma pyörii hyvin myös Netbeansin kautta. Ohjelman netbeans projekti löytyy Githubista ilmoitetusta repostista. Ohjelmassa testataan eri tietorakenteita, jotka toteuttavat joitain joukko-operaatioita. Tietorakenteet ovat map tyyliä eli joka solmussa on sekä key ja value.

Ohjelman konsolissa komennot annetaan antamalla ensin tietorakenteen nimi johon toimenpide kohdistuu tämän jälkeen ajettavan metodin nimi, tämän jälkeen haluttaessa parametri/parametrit joilla metodi ajetaan. Parametrien antaminen ei ole pakollista, jos parametreja ei anneta ajetaan metodi arvotuilla parametreilla. Viimeisenä annetaan toistojen määrä. Kaikki komennon osat erotetaan toisistaan välilyönnillä. Komennon eteen ei saa laittaa ylimääräisiä välilyöntejä.

Tietorakenteen_nimi metodin_nimi parametri/parametrit(vaihtoehtoinen) toistot
Käskyn rakenne.

AVL_Tree add_identical 101 1

Esimerkki kysely avl-puulle. Metodi add_identical lisää solmun jossa sekä key ja value ovat 101. Toistoja yksi kerta.

AVL_Tree add_identical 20000

Avl-puulle lisätään 20000 solmua. Komennossa ei annettu parametreja, jolloin ohjelma arpoo parametrit. Toistojen määräksi annettiin 20000

Ohjelma tulostaa kauanko aikaa komennon suorittamiseen kului ja mitä metodi palautti viimeisellä palautus kerralla. Suoritettaessa ensimmäinen lisäys komento tietorakenteelle on komennon suoritus huomattavasti hitaampi kuin ensimmäisellä kerralla. Tietorakenteet ovat erillisiä eli yhteen tietorakenteeseen lisääminen ei lisää toiseen tietorakenteeseen mitään. Tehdyt komennot eivät nollaudu automaattisesti (ohjelma ei tallenna mitään eli uudelleen käynnistettäessä kaikki tietorakenteet nollataan). Ohjelman aikana tietorakenne voidaan nollata helposti *clear* komennolla. Kutakin tietorakennetta on ohjelmassa käytössä yksi kappale.

Jos parametreja ei anneta ohjelma antaa parametriksi luvut nollasta toistojen määrä – 1:teen. Numerot ovat satunnaisessa järjestyksessä. Tästä johtuu kuitenkin esimerkiksi se että jos poistettaessa joukosta ensin 200 solmua poistuvat solmut 0 – 199. Toisella kerralla poistettaessa 200 solmua yrittää ohjelma uudestaan poistaa solmut 0 – 199 eikä koko pienene (saman tapainen efekti esimerkiksi lisäämisessä).

Ohjelmassa on kaksi apu komentoa *Available data structures?* joka luettelee kaikki käytössä olevat tietorakenteet. Luettelossa on mukana myös paketti jossa tietorakenne sijaitsee. Paketin kertovaa osaa ei käytetä komennossa.

Komentoa *add_identical* kannattaa käyttää jos haluaa että sekä key ja value solmuissa ovat samat. Tämä on usein haluttua jos halutaan tutkia min, max, successor tai predecessor operaatioita. Tietorakenne voidaan helposti nollata *clear* metodilla ja antamalla toistojenmääräksi yksi. Tietorakenteen solmujen määrä saadaan myös helposti metodilla *size* ja antamalla toistojen määräksi yksi. Kaikki metodit ottavat parametreikseen kokonaislukuja.

Ohjelmassa käytössä olevat metodit ja parametri määrät:

<u>Metodi</u>	<u>Parametreja</u>
add	2
remove	1
get	1
min	0
max	0
clear	0
contains	1
size	0
predecessor	1
contains_key	1
successor	1
add_identical	1

HUOM! Katso mitä metodit tekevät JavaDocista. Esimerkiksi kaikki listan ja hash_mapin metodit eivät tee välttämättä täysin samaa asiaa kuin puiden vastaavat metodit.

HUOM! Ohjelmaa ei kannata laittaa suorittamaan miljoonilla toisto kerroilla, sillä siihen saattaa kulua huomattavan paljon aikaa. Lisäksi lisäämällä riittävästi solmuja saattaa muisti loppua kesken.