

Toteutusdokumentti

Ohjelman yleisrakenne

Ohjelman tausta osa (sitsiplaseeraus.random ja Lataaja)

- tuottaa satunnaiset sitsaajat, heidän sukupuolet, avecit ja mieltymykset muista sitsaajista.
- kirjoittaa ja lukee asetustiedostoja.
- tulostaa yhteydet ja sitsaajat löydettyssä järjestyksessä.

omatTietorakenteet

- Toteuttaa ArrayListin toiminnallisuuden mahdollisimman tehokkaasti ja helposti vain siltä osin, kuin muulle ohjelmalle tarpeellista.
- Toteuttaa Hakemisto, joka on ulkoisesti lähes kuin javan oma HashMap, mutta sisäisesti vain avain-arvolla laajennettu ArrayList.
 - Luokka Vektori, joka mahdollistaa Hakemiston foreach läpikäymisen.

Sitsiplaseeraus

- Luokat toimivat eräänlaisina tiedon varastoina, jotka käyttävät melko vähän mitään kovin hienoa logiikkaa.
 - Sitsaaja sisältää sitsaajan tiedot ja mieltymykset
 - Paikka vastaa hieman reaali maailman tuolia, joka tietää paikkansa ja niin kerrottaessa myös kuka hänen paikallaan istuu. Tietää oman paikkansa ja viereiset tuolit.
 - Sitsit pitää kaiken nämä koossa, ja tietää pöytien määrän ja järjestyksen pöydissä.

Pisteyttäjä

- PaikanPisteet tietää ketä hän seuraa, ja osaa laskea paikalla istuvan henkilön pisteet.
- Pisteet tietää kaikki paikat ja laskee paikkojen pisteet yhteen.
- Sisältää auttavan luokan Laskin, jonka virka on pitää muistissa jo ennestään lasketut etäisyydet vastakkaiselle puolelle pöytää (hypot).

Järjestelijä

- Järjestäjä-luokka vaihtaa joko sille annettujen paikat keskenään tai arpoo kaksi satunnaista paikkaa ja vaihtaa niiden sitsaajat keskenään.
- Optimoija luo ensiksi hyvä tyttöpoika-järjestyksen avecit ja puoliset huomioonottaen, jonka jälkeen pyrkii järjestämään Järjestäjää-hyväksikäyttäen myös mieltymysten mukaisesti niin, että koko sitsien pistemäärä kasvaa koko ajan.
- ParhaanLöytäjä kutsuu optimoijaa aina uudestaan sen löydettyä siihen mennessä parhaan, ja tallentaa parhaan siihen mennessä löydetyn järjestyksen muistiin.

Saavutetut aika- ja tilavaativuudet

Ongelma ongelman ratkaisussa oli, ettei parasta mahdollista järjestystä ole helppo löytää. Ohjelma löytyy kuitenkin muutamissa sekunneissa huomattavasti paremman järjestyksen kuin ihminen pystyisi.

Ohjelma kuitenkin löytää melko nopeasti erittäin lähellä hyvää olevan järjestyksen suurillakin sitsaaja määrillä, ja muutamissa kymmenissä sekunneissa jo lähes täydellisyyttä hipovan.

Ohjelma käyttää noin 20MB muistia lähes riippumatta sitsien kooska, josta varmasti suurin osa menee Javan omaan kalustoon.

Suorituskykyvertailu

Joitain tuloksia:

- 12hlö kolmessa pöydässä, keskimäärin 0.12sec
- 18hlö kolmessa pöydässä, keskimäärin 0.29sec
- 24hlö kolmessa pöydässä, keskimäärin 16.56sec

Jo 60hlö sitseillä sadatkaan sekunnit eivät riitä löytämään parasta mahdollista, joten tulosten vertailu keskenään on hieman hankalaa.

Ohjelma käy kuitenkin läpi noin 2000 järjestystä ja pisteiden laskua sekunnissa. Tähän vaikutti huomattavasti siirtyminen HashMapista Hakemistoon, sillä HashMapin foreach-oli erittäin suuri pullonkaula.

Työn mahdolliset puutteet ja parannusehdotukset

Parempiin tuloksiin pääsee, jos osaa (ja ymmärtäisi) vaihdella seuraavia muuttujia asetustiedoston ja tilanteen mukaan.

- Kuinka monta vaihtoa yritetään tehdä, ennen kuin lasketaan pisteet.
 - Jos pieni, niin nopeampi, koska todennäköisemmin tekee onnistuneen siirron kuin epäonnistuneen (käytännössä todettu).
- Missä vaiheessa käydään kaikki yhden siirron päässä olevat järjestyksen läpi yksi kerrallaan (löytää parempia järjestyksiä helposti ja nopeasti, mutta tuottaa (ehkä) liian helposti samanlaisia tuloksia edellisten kanssa, koska ei satunnaisuutta)
- tyttöpoika-järjestyksen parempi ja tehokkaampi luominen, sillä tämä oli suuri tekijä, kun samaa sukupuolta olevat tekivät hankalaksi löytää hyvää järjestystä.

Asetustiedoston luontiin pitää luoda jonkinlainen käyttöliittymä, mutta tämä ei liity tähän kurssiin :)

Lähteet

- <http://docs.oracle.com/javase/7/docs/api/>
- <http://stackoverflow.com/>