TKT20010 Aineopintojen harjoitustyö: Tietorakenteet ja algoritmit

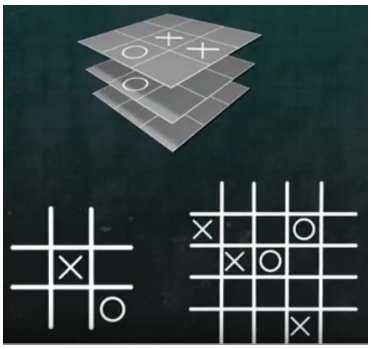
Kevät 2018 (Periodi 3)

Määrittelydokumentti:	Ristinolla3D
Github-alikansio:	/KO-Ristinolla
Tekijän nimi:	Kari Ojala
Päiväys:	19.1.2018

1. Yleiskuvaus

Tehtävänä on laatia tekoäly, peliohjelma, joka pelaa kolmiulotteista ristinollaa. Ohjelma voi pelata joko itseään vastaan tai ihmistä vastaan. Ohjelma laaditaan Java-kielellä.

Useimmille on tuttu ristinolla-pelin alkeellinen versio, jossa on 2-ulotteinen (2D) 3x3 ruudukko (Kuva 1 vasen alakulma). Pelin alussa ruudukko on tyhjä. Pelissä on kaksi pelaajaa, jotka vuorotellen merkkaavat ruudukkoon oman merkkinsä, ristin tai nollan. Pelin voittaa se, joka ensimmäisenä saa muodostettua kolmen peräkkäisen merkin linjan joko vaakasuoraan, pystysuoraan tai diagonaaliin.


Kuva 1[1].

Peliä voidaan laajentaa monin tavoin. Voidaan kasvattaa ruudukon kokoa tai lisätä ulottuvuuksia[2]. Kuvassa keskellä ylhäällä on 3-ulotteinen (3D) 3x3x3-ruudukko. Siinä voittolinjoiksi kelpaavat kaikki kolmen ”koordinaattiakselin” suuntaiset linjat ja kunkin tason viistolinjat sekä koko kuution lävistävät viistolinjat. (3x3x3-peliä voidaan vaikeuttaa poistamalla siitä keskimmäisen tason keskellä oleva ruutu[5], koska yksinkertaisin voittava strategia sattuu olemaan se, että aloittaa tästä ruudusta. Tätä voidaan ohjelmalla tutkia, jos se ei aiheuta kohtuutonta lisätyötä. Tämä on myös keino pienentää aikavaativuutta tarvittaessa.)

Tässä työssä tarkastellaan pelejä, joiden koko on 3x3x3 (ja mahdollisesti 4x4x4). PBS:n videolla[1] on kuva, joka esittää, mistä laajennuksista on todistetusti löydetty voittava tai tasapelin tuottava pelistrategia.

Kuva 2. Pelin aloittajalle voittavat (W) ja tasapelin (D) tuottavat strategiat eri kokoisilla ristinolla-ruudukoilla. Pystyakselilla pelin dimensio (d) , vaaka-akselilla ruudukon leveys (w). Kysymysmerkki ruudussa tarkoittaa, ettei kyseistä tulosta ole vielä matemaattisesti todistettu eikä tulos ole edes täysin varma.

2. Algoritmit ja tietorakenteet

Perusoletus on, että 3x3x3-peli on ratkaistavissa modifioimalla Tietorakenteet ja algoritmit kurssimonisteen[3] sivuilla 394-400 esitettyä pelipuu-menetelmää. Koska pelin aikavaativuus kasvaa erittäin jyrkästi suhteessa 3D-ruudukon kokoon, on epävarmaa, löytyykö edes 3x3x3-pelille kohtuullisessa ajassa ratkaistavaa algoritmia. Jos löytyy, kokeillaan sitä 4x4x4-pelille. Verkosta löytyy toinen kiinnostava versio[5].

Pelipuu pyritään ensisijaisesti toteuttamaan rekursiivisella ja syvyyssuuntaisella minimax-algoritmilla. Koska perinteinen minimax-algoritmi käy läpi koko pelipuun, se ei ehkä sellaisenaan sovi edes 3x3x3-pelin ratkaisemiseen, jos sen aikavaativuus on O(n!), missä n on alkeiskuutioiden lukumäärä. (3x3x3-ruudukon tapauksessa on n=27.) Aikavaativuutta on mahdollista alentaa, jos löytyy tehokkaita tapoja välttää esim. symmetristen puun haarojen läpikäynti. Lisäksi peliruudukko/kuutio itsessään on symmetrinen. Tekoäly-kurssin kurssimateriaalissa[4] mainitaan Alpha-beta-karsinta toisena keinona, jolla oikea minimax-päätös voitaisiin tehdä käymättä läpi kaikkia pelipuun solmuja. Jos pelipuut voidaan käytännössä evaluoida vain tiettyyn syvyyteen asti, voidaan kokeilla ja verrata muutamia erilaisia pelitilanteiden arviointitapoja.

Pelipuun perustietorakenteena on 3x3x3 –taulukko (tai kolme 3x3-taulukkoa). Javan TreeSet-tietorakenne, jos se sallitaan, saattaa sopia aputietorakenteeksi listaamaan pelitilanteita joita ei kannata tutkia, mikäli sen käyttö on sallittu. (Allekirjoittanut käytti TreeSet-rakennetta mm. Tietorakenteet ja algoritmit -kurssin Sudoku-tehtävän tehokkaassa ratkaisussa.)

3. Ohjelman syötteet ja tulosteet

Ohjelma pyytää tarvittavat syötteet käyttäjältä yksinkertaisten valikkojen avulla. Siirrot voidaan antaa numerojonona muodossa KLM, jossa kukin kirjain vastaa yhtä nxnxn-taulukon koordinaattia. Jokaisen siirron jälkeen pelitilanne tulostetaan kuvaruudulle. 3x3x3 –taulukon tapauksessa tulostetaan kolme 3x3 –taulukkoa alekkain. Taulukoiden viereen tulostetaan yksinkertainen ohje koordinaattien antamisesta.

Ohjelma tarkistaa syöttötiedon kelpoisuuden ja pyytää tarvittaessa uuden syötteen. Mikäli kone asetetaan pelaamaan itseään vastaan, käyttäjä voi halutessaan tulostaa siirrot yksi kerrallaan. Lisäksi voidaan tallentaa aputaulukoihin tai tiedostoihin erilaista aika- ja tilastollista dataa pelin kulusta.

4. Aika- ja tilavaativuudet

Pahimmassa tapauksessa n alkeiskuution 3D-ristinollapelin aikavaativuus on O(n!). 3x3x3-ristinollassa n=27: ensimmäinen siirto voidaan siis valita 3*3*3=27 eri tavalla, seuraava 26 eri tavalla jne.

Alustava tilavaativuus on luokkaa O(n), missä n on taulukon koko, esim. 3x3x3. Jos rekursiivisia kutsuja on enimmillään yhtä monta kuin peliruudukossa alkeiskuutioita ja jokaiselle tasolle tarvitaan yksi taulukko, tilavaativuus kasvaa kertoimella n. Tilavaativuus on siten pahimmillaan $O(n^2)$.

5. Lähteet

1. PBS:n video, Higher-Dimensional Tic-Tac-Toe, https://www.youtube.com/watch?v=FwJZa-helig .
2. Tic-tac-toe_variants, Wikipedia-artikkeli, https://en.wikipedia.org/wiki/Tic-tac-toe_variants .
3. Kivinen, J., Tietorakenteet ja algoritmit kurssimoniste, syksy 2017, ss. 394-400.
4. Roos, T., Johdatus tekoälyyn, 2016, ss. 8-12.
5. Eräs modifioitu 3x3x3 ristinolla-peli http://www.myhip.com/ttt3d/3d_tic_tac_toe_too.html .