Laskuharjoitus 4

7.2.2002

1. Kuvaile peräytymishakualgoritmit seuraaville ongelmille:

   (a) Etsi kaikki tavat asettaa n kuningatarta $n \times n$-shakkilaudalle siten, etteivät kuningattaret uhkaa toisiaan.

   (b) Etsi $n \times n$ miinaharavapelin (minesweeper) annetusta peliasemasta ruudut

      i. joissa ei varmasti ole miinaa; ja
      ii. joissa on varmasti miina.

   (Miinojen lukumäärä oletetaan tuntemattomaksi.)

2. Määritellään graafin $\mathcal{G}$ kromaattinen luku

$$\chi(\mathcal{G}) = \min\{k \ : \ \mathcal{G}\text{:lla on solmujen } k\text{-väritys}\}$$

   ja klikkiluku

$$\omega(\mathcal{G}) = \max\{k \ : \ \mathcal{G}\text{:ssä on } k\text{:n kokoinen klikki}\}.$$

   Kirjan lemman 4.4 mukaan $\omega(\mathcal{G}) \leq \chi(\mathcal{G})$. Osoita, että epäyhtälö voi olla myös aito: etsi graafi, jolle $\omega(\mathcal{G}) < \chi(\mathcal{G})$.

3. Olkoon annettuna graafi $\mathcal{G}$ ja sen solmuille jokin järjestys. Ahne graafinväritysalgoritmi värittää graafin solmu kerrallaan pienimmästä suurimpaan siten, että kulloinkin väritettävän solmun väriksi valitaan pienin väri (oletetaan värit järjestetyiksi), joka ei esiinny väritettävän solmun jo väritetyissä naapureissa.

   (a) Esitä ääretön perhe graafeja, joiden solmut on järjestetty siten, että ahne väritysalgoritmi ei tuota optimaalista väritystä. (Väritys on optimaalinen jos värien lukumäärä on $\chi(\mathcal{G})$.)

   (b) Onko kaikille graafeille olemassa sellainen solmujen järjestys, että ahne algoritmi tuottaa optimaalisen värityksen?

Laskuharjoituksen 4 vastaukset

Tehtävä 1 (a)

Oletetaan shakkipelin säännöt tunnetuiksi. Tarkastellaan kutakin laudan riviä $1, \ldots, n$ vuorollaan. Valintajoukkona on tällöin niiden tällä rivillä olevien ruutujen joukko, joita ei vielä uhkaa laudalla oleva kuningatar. Kokeillaan kutakin näistä vuorollaan ja siirrytään seuraavalle riville.

Rivejä ei välttämättä kannata käydä numerojärjestyksessä läpi. Tehokkaampi vaihtoehto on järjestää haku niin, että kulloinkin tarkasteltavaksi riviksi valitaan se, jolla sillä hetkellä on vähiten uhkaamattomia ruutuja. Tällöin haun haarautumiskerroin lähellä hakupuun juurta on pieni, jolloin hakusolmujen määrä pienenee, vaikka kaikki vaihtoehdot käydäänkin yhtä kattavasti läpi.

Tehtävä 1 (b)

Miinaharavapelissä pelaajan tehtävänä on löytää annetulta $n \times n$ pelilaudalta kaikki ruudut, jotka sisältävät miinan. Pelaajan käytettävissä on miinaharava, joka ilmoittaa pelilaudan ruudussa miinojen lukumäärän kyseistä ruutua ympäröivissä ruuduissa. Pelin peliasema koostuu joukosta tunnettuja ruutuja, joissa tiedetään miinaharavan lukema. Peli etenee asemasta toiseen siten, että pelaaja osoittaa laudalta tuntemattoman ruudun. Jos ruudussa oli miina, peli päättyy. Jos ruudussa ei ollut miinaa, miinaharavan uusi lukema liitetään peliasemaan, ja peli jatkuu. Peli päättyy pelaajan voittoon, kun kaikissa tuntemattomissa ruuduissa on miina. (Tässä pelin versiossa miinojen lukumäärä oletetaan tuntemattomaksi.)

Olkoon pelilaudan ruutujen joukko $D_n = \{1, 2, \ldots, n\} \times \{1, 2, \ldots, n\}$. Ruudun (x, y) *ympäristö* on joukko

$$N(x, y) = \{(x, y) + (\delta_1, \delta_2) \ : \ -1 \leq \delta_1, \delta_2 \leq 1, \ (\delta_1, \delta_2) \neq (0, 0)\} \cap D_n.$$

Huomaa, että $(x, y) \notin N(x, y)$.

Olkoon $\mathsf{mines} \subseteq D_n$ niiden ruutujen joukko, joissa on miina. (Tämä joukko on luonnollisesti pelaajalle tuntematon.) Miinaharavan lukema ruudussa (x, y) on tällöin

$$d(x, y) = |\mathsf{mines} \cap N(x, y)|.$$

Olkoon $\mathsf{known} \subseteq D_n$ pelaajan osoittamien ruutujen joukko. Jos peli ei ole päättynyt, pätee $\mathsf{known} \cap \mathsf{mines} = \emptyset$. Oletetaan tämä todeksi.

Koska miinojen kokonaismäärä on tuntematon, pelaaja voi saada tietoa miinojen lukumäärästä ja sijainnista vain miinaharavan tunnetussa alueessa antamien lukemien avulla. Määritelmän mukaan miinaharavan lukema ruudussa

(x, y) antaa tietoa vain ruutujen $N(x, y)$ sisältämistä miinoista, joten pelaaja voi päätellä miinojen sijainnista jotain vain tunnetun alueen *reuna-alueella*

$$\mathsf{boundary} = \big(\bigcup_{(x,y)\in \mathsf{known}} N(x, y)\big) \setminus \mathsf{known}.$$

Reuna-alueen ulkopuolisista tuntemattomista ruuduista ei voida päätellä mitään. (Huomaa, että edellinen väite ei pidä paikkaansa jos miinojen lukumäärä on tunnettu!)

Rajoitetaan tarkastelu reuna-alueeseen $\mathsf{boundary}$. Sanotaan joukkoa

$$E \subseteq \mathsf{boundary}$$

konsistentiksi, jos kaikilla $(x, y) \in \mathsf{known}$ pätee $d(x, y) = |E \cap N(x, y)|$. Tällöin joukko on konsistentti jos ja vain jos on mahdollista, että miinat todellisuudessa sijaitsevat joukon E ruuduissa. Erityisesti, joukko $\mathsf{mines} \cap \mathsf{boundary}$ on konsistentti.

Olkoon $\mathscr{E}$ kaikkien konsistenttien joukkojen joukko. Määritellään joukot

$$\mathsf{safe} = \mathsf{boundary} \setminus \bigcup_{E\in\mathscr{E}} E, \quad \text{ja} \quad \mathsf{flag} = \bigcap_{E\in\mathscr{E}} E.$$

Tarkastellaan ensin joukkoa safe. Nyt $\mathsf{safe} \cap \mathsf{mines} = \emptyset$, koska $\mathsf{mines} \cap \mathsf{boundary} \in \mathscr{E}$. Joukko safe on siis joukko turvallisia ruutuja, joissa ei varmasti ole miinaa. Toisaalta safe on myös suurin mahdollinen joukko jolle edellinen väite voidaan esittää, koska kaikki pisteet $x \in \mathsf{boundary} \setminus \mathsf{safe}$ sisältyvät johonkin konsistenttiin joukkoon E, ja miinaharavan lukemien perusteella ei voida sulkea vaihtoehtoa $\mathsf{mines} \cap \mathsf{boundary} = E$ pois.

Tarkastellaan seuraavaksi joukkoa flag. Koska $\mathsf{flag} \subseteq E$ kaikilla $E \in \mathscr{E}$, pätee tämä erityisesti joukolle $E = \mathsf{mines} \cap \mathsf{boundary}$. Joten, $\mathsf{flag} \subseteq \mathsf{mines}$. Joukko flag sisältää siis vain ruutuja, joissa on miina. Toisaalta kaikille pisteille $x \in \mathsf{boundary} \setminus \mathsf{flag}$ on olemassa konsistentti joukko E, jolle $x \notin E$. Näin ollen flag on suurin joukko, jonka ruuduissa on varmasti miina.

Joukot safe ja flag ovat siis täsmälleen tehtävän (i) ja (ii)-kohdissa kysytyt joukot.

Peräytyvää hakua voidaan käyttää joukkojen safe ja flag määrittämiseen esimerkiksi seuraavasti:

- Laaditaan peräytyvä hakualgoritmi, joka tuottaa ratkaisuina kaikki konsistentit joukot.
- Ennen hakua alustetaan $\mathsf{safe} \leftarrow \mathsf{boundary}$ ja $\mathsf{flag} \leftarrow \mathsf{boundary}$.

- Käydään peräytyvällä haulla läpi kaikki konsistentit joukot E. Aina konsistentin joukon löydyttyä haun aikana asetetaan

$$\mathsf{safe} \leftarrow \mathsf{safe} \setminus E \quad \text{ja} \quad \mathsf{flag} \leftarrow \mathsf{flag} \cap E.$$

- Kun kaikki konsistentit joukot on käyty läpi, joukot safe ja flag ovat halutut joukot. (Huomaa, että molemmat joukot voivat olla tyhjiä. Tämä tapahtuu esimerkiksi pelin alkutilanteessa, jolloin known $= \emptyset$.)

Konsistentit joukot voidaan tuottaa peräytyvällä haulla esimerkiksi seuraavasti.

- Olkoon joukossa boundary N alkiota $P_1, \ldots, P_N$, $N \geq 1$. (Erikoistapaus $N = 0$ jätetään huomiotta.)
- Mallinnetaan joukon boundary osajoukkoja $E \subseteq$ boundary binäärilistoina $\vec{E} = [x_1, \ldots, x_N]$, joille $x_i = 1 \Leftrightarrow P_i \in E$ ja $x_i = 0 \Leftrightarrow P_i \notin E$.
- Osittaisratkaisu on lista $[x_1, \ldots, x_{k-1}]$; alussa tyhjä lista.
- Hakupuussa syvyydellä k lisätään listaan vuorotellen alkiot $x_k = 0$ ja $x_k = 1$, ja jatketaan hakua eteenpäin.
- Kun listassa on N alkiota, tarkistetaan, onko sitä vastaava joukko konsistentti. (Jos on, päivitetään joukkoja safe ja flag kuten edellä.) Tämän jälkeen palataan aiemmalle tasolle hakupuussa.
- Hakua voidaan *huomattavasti* tehostaa karsimalla sellaisia osittaisratkaisuja, joita ei selvästi voida täydentää konsistenteiksi joukoiksi. Palataan tähän tuonnenpana.

Tehtävä 2

Viiden solmun sykli $\mathcal{C}_5$ (viisikulmio) on kromaattiselta luvultaan 3, mutta sen suurimman klikin koko on 2.

Tehtävä 3 (a)

Olkoon $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ graafi, ja $\mathcal{W} \subseteq \mathcal{V}$. Solmujoukko $\mathcal{W}$ on *riippumaton* (independent), jos minkään sen kahden solmun välillä ei ole kaarta. Graafi $\mathcal{G}$ on *kaksijakoinen* (bipartite) jos sen solmujoukko $\mathcal{V}$ voidaan jakaa kahteen ei-tyhjään riippumattomaan joukkoon $\mathcal{W}$ ja $\mathcal{V} - \mathcal{W}$.

Kaksijakoinen graafi voidaan aina värittää kahdella värillä, koska molempiin riippumattomiin joukkoihin riittää yksi väri.

Täydellinen kaksijakoinen (complete bipartite) graafi $(\mathcal{V}, \mathcal{E})$ sisältää kaikki mahdolliset kaaret sen kahden riippumattoman joukon välillä, ts. kaikilla $u \in \mathcal{W}$ ja kaikilla $v \in \mathcal{V} - \mathcal{W}$ pätee $\{u, v\} \in \mathcal{E}$.

Konstruoidaan graafiperhe seuraavasti: Otetaan täydellinen kaksijakoinen graafi, jonka molemmissa riippumattomissa joukoissa on n solmua, ja järjestetään solmut siten, että ensimmäinen riippumaton joukko koostuu parittomista solmuista $\{1, 3, \ldots, 2n-1\}$, ja toinen parillisista $\{2, 4, \ldots, 2n\}$. Poistetaan graafista kaaret $\{1, 2\}$, $\{3, 4\}$, $\{5, 6\}, \ldots, \{2n-1, 2n\}$.

Ahne algoritmi värittää graafin seuraavasti: Solmulle 1 annetaan väri 1. Solmulle 2 annetaan väri 1, koska graafissa ei ole kaarta $\{1, 2\}$. Solmulle 3 annetaan väri 2, koska graafissa on kaari $\{2, 3\}$, ja väriä 1 ei näin ollen voi käyttää. Solmulle 4 annetaan väri 2, koska kaaren $\{1, 4\}$ takia väriä 1 ei voi käyttää. Solmulle 5 annetaan väri 3, koska kaarien $\{2, 5\}$ ja $\{4, 5\}$ takia värejä 1 ja 2 ei voi käyttää. Solmulle 6 annetaan väri 3, koska kaarien $\{1, 6\}$ ja $\{3, 6\}$ takia värejä 1 ja 2 ei voi käyttää, jne.

Yleisessä tapauksessa pariton solmu $2k-1$ $(k \geq 2)$ väritetään värillä k, koska kaaret $\{2j, 2k-1\}$, $1 \leq j \leq k-1$ estävät k:ta pienempien värien käytön, sillä solmu $2j$ on väritetty värillä j kaikilla $1 \leq j \leq k-1$. Toisaalta parillinen solmu $2k$ $(k \geq 2)$ väritetään värillä k, koska kaaret $\{2j-1, 2k\}$, $1 \leq j \leq k-1$ estävät k:ta pienempien värien käytön, sillä solmu $2j-1$ on väritetty värillä j kaikilla $1 \leq j \leq k-1$.

Graafin ahneeseen väritykseen tarvitaan siis lopulta n väriä, vaikka graafi on väritettävissä ainoastaan kahdella värillä.

Tehtävä 3 (b)

On selvää, että värityksestä riippumatta samalla värillä väritetyt graafin solmut muodostavat riippumattoman joukon. Oletetaan, että käytettävissä on optimaalinen väritys. Jos graafin solmut nyt listataan väriluokka kerrallaan (esim. ensin kaikki punaiset solmut, sitten siniset, jne.), saadaan solmuille järjestys, jolla ahne algoritmi tuottaa optimaalisen värityksen. Saatu ahne väritys ei välttämättä ole identtinen alkuperäisen värityksen kanssa, mutta värien lukumäärä on sama.