
Diskreetti matematiikka, syksy 2010

Matlab-harjoitus 4 (2.12. klo 16-18 MP102, osa jää varmaan bonusetähtäväksi)

Tehtäviin vastataan tälle paperille, osoitettuihin tyhjiin alueisiin, yleensä tyhjille riveille. Tehtävät saa – ja on suositeltavaa – ratkaista parityöskentelynä.

Varmista, että verkkolevyysi **X**: juuressa (!) on kansio nimeltä **DISKREET**. Lataa (klikkaa tms. riippuen selaimesta) Kurssimateriaalisivulta **Diskreet3.zip** ja pura se levyysi **X**: juureen. Sisälön pitäisi mennä tuohon kansioon **X:\DISKREET**. Käynnistä Matlab ja siirry siellä em. kansioon. Aja nyt M-skriptit **alustus** ja **ston**. Nyt sinun pitäisi olla kansiossa **X:\DISKREET\VERKOT\SUUNTAAM**

1. Suuntaamattoman verkon generointi ja piirto

a) Suorita ajatuksella seuraavat käskyrivit:

```
» n = 5; Ma = ones(n) - eye(n);
```

Mitä tässä tehtiin (verkkoteoreettisesti)? _____

b) Suorita edelleen käskyrivit:

```
» v = [3,4], p = v(1), q = v(2),  
» n = p + q  
» Mb = ones(n);  
» Mb(1:p,1:p) = 0  
» Mb((p+1):n,(p+1):n) = 0
```

Mitäs tässä? _____

c) Piirrä kohdissa a) ja b) konstruoidut verkot. Käytettävissäsi on piirtofunktio **PiirSTon**. Mitä erikoista on matriisiin **Mb** liittyvässä verkossa? _____

d) Selvitä suuntaamattomia satunnaisverkkoja generoivien funktioiden **SatVerk** ja **NSatVerk** toimintaa; mitä pakollisia ja optionaalisia parametreja tarvitaan, mikä on niiden vaikutus lopputulokseen.

SatVerk: _____

NSatVerk: _____

2. Matriisiaritmetiikkaa – miksi?

```
» M = SatVerk(5,2); M(1,1) = 0  
» M + eye(5)  
» K = ones(5) - sign(M + eye(5))  
» PiirSTon(M), PiirSTon(K,'-b')
```

Mitäpä tehtiin edellä? _____

Kirjoita funktio, joka tekee edellä kuvatun toimen.

3. Verkon syöttö ja muunnokset

a) Suunnittele ensin paperilla kuusisolmuinen verkko solmuina $\{1, 2, 3, 4, 5, 6\}$, joka jakaantuu tasan kahteen osaan $\{1, 2, 3\}$ ja $\{4, 5, 6\}$ niin, että osista ei ole reittejä toisiinsa, mutta osien sisällä esiintyy rinnakkaisiakin kaaria. Sen jälkeen syötä verkko vuorovaikutteisella Matlab-funktiolla **Syotto**, joka (yrittää) muodostaa verkon matriisin. Aloita vaikkapa näin

```

» type Syotto
» Ma = Syotto(' ')
Montako solmua verkossa on? 6
Solmusta 1 solmuihin: [1,3]
etc.

```

b) Jos ohjelma herjaa, että matriisista ei tullut symmetrinen (tarkemmin sanottuna herjaa on funktio Tarkastu, joka tsekkaa, kelpaako matriisi suuntaamattoman verkon matriisiksi), voit noudattaa mukana tulevaa vihjettä ja käyttää funktiota SymSoi. Piirrä verkkosi ja katso tuliko se mitä tarkoitt!

Mitä tekee funktio SymSoi ja miten? _____

Mitä tekee funktio Listaksi? _____

4. Verkon yhtenäisyys a) Kopioi matriisiksi M seuraava:

```

0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0
0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 1 0 0
1 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0
1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0
0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0
0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0
0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1
0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 1 0 0
0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1
1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0
0 1 1 0 0 0 0 0 0 0 1 0 0 0 1 1 1 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1
1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 1 0

```

b) Suorita taas ajatuksella seuraavat:

```

» M = sign(M - diag(diag(M)))
» PiirSTon(M)
» uudet = logical(M(1,:))
» solmut = find(uudet)
» for i = solmut, PiirKaar(M,1,[1 i], '--w'), end; PiirSolm(M,solmut,'*w')
» mukana = uudet
» mukana(1) = 1;
» [M(uudet,:); mukana]
» vanhat = mukana
» mukana = any([M(uudet,:); mukana])
» uudet = mukana & ~vanhat
» solmut = find(uudet)
» for i = solmut, PiirKaar(M,1,[1 i], '--y'), end; PiirSolm(M,solmut,'*y')
» vanhat = mukana
» mukana = any([M(uudet,:); mukana])
» uudet = mukana & ~vanhat

```

toistaen ja vaihtaen piirtoväriä kunnes uudet on pelkkää nollaa.

Mitä edellä tehtiin? _____

Mitä kuvaa vektori `uudet`? _____

Mikä on sopiva lopetusehto, kun `yo` koodi tehdään silmukaksi? _____

5. Tee MATLAB-funktio `YhtenKo`, joka saatuaan suuntaamattoman verkon matriisiin M palauttaa arvon 1, jos verkko on yhtenäinen, muutoin arvon 0.

Vihje: Käytä *breadth-first*-menetelmää, jossa voi hyödyntää Matlabin näppärää matriisi-manipulointia (ks. tehtävä 4):

valitaan lähtösolmuksi 1 ja katsotaan mihin solmuihin siitä pääsee yhdellä kaarella, mihin puolestaan niistä, jne. Kussakin vaiheessa lisätään kyseiset solmut *solmuindikaattoriin* mukana, joka on solmumäärän pituinen 0-1-vektori ja jolle annetaan aluksi arvo $M(1, :)$. Tähän vaihdetaan ykkösiä sitä mukaa, kun solmuissa on vierailtu. Prosessi päättyy, kun tulee tilanne, ettei uusiin solmuihin enää päästä. Jos mukana sisältää vain ykkösiä, on verkko yhtenäinen. Testaa funktioitasi (käytä apuna satunnaisverkko- ja piirtofunktioita).

6. Suuntaamattoman verkon alkeiskäsittelyä

- a) Ota käyttöön esimerkkiverkkoja lataamalla tiedosto `verkot.mat`, kirjoittamalla

```
» load verkot
```

```
» who
```

Suorita ajatuksella seuraavat käskyrivit:

```
» M = erill3
```

```
» PiirSTon(M) % sovita molemmat ikkunat esille
```

```
» any(M)
```

```
» ~any(M)
```

```
» find(~any(M))
```

Mitä tässä tehtiin: _____

- b) Suorita edelleen käskyrivit:

```
» M
```

```
» PiirSTon(M) % jos ei ole jo
```

```
» sum(M)
```

```
» diag(M)
```

```
» diag(M).'
```

```
» sum(M) + diag(M).'
```

Mitäs tässä: _____

7. Kirjoita Matlab-funktio

- a) `Erilliset`, joka etsii verkon erilliset solmut, ja palauttaa ne vektorina.

```
function ER = Erilliset(M); % etc
```

- b) `Asteet`, joka laskee ja antaa ulos solmujen asteluvut. Muoto vaikkapa kaksirivisenä matriisina, jossa ylempi rivi on solmujen numerot ja alempi niiden asteluvut. Voit käyttää rivivektorien x ja y yhdistämiseen tyyliä $[x ; y]$.

```
function AST = Asteet(M); % etc
```

8. a) Suorita ajatuksella seuraavat käskyrivit:

```
» rem([3,2,6,5,8,7,4,5,6],2)
» help rem
» M = nsatverk(8,1.5)
» PiirSTon(M)
» rem(sum(M),2)
```

Mitä tässä tehtiin: _____

Diagonaali muodostaa ongelman. Mutta diagonaalilla oleva luku lasketaan astelukuun kaksinkertaisena, joten asteluvun parittomuuteen se ei vaikuta. Tässä yhteydessä voidaan siis vaikka nollata diagonaali!

```
» MD = M - diag(diag(M))
» rem(sum(MD),2)
» Asteet(MD)
» find(rem(sum(MD),2))
```

Mitäs tässä sitten saatiin aikaan? _____

b) Tee MATLAB-funktio `PTonAst`, joka saatuaan suuntaamattoman verkon matriisiin M tulostaa paritonasteiset solmut vaakavektorina (joka voi olla tyhjä). Voit esimerkiksi aloittaa nollaamalla diagonaalin, joka sotkee asteluvun laskemista. Siis:

```
function PTAST = PTonAst(M);
M = M - diag(diag(M)); % etc
```

9. Yhdistä edellä tehdyt osaset funktioksi `EulerKo`, joka tarkastaa onko annetun verkon Euler-ominaisuudet. Tarkemmin sanottuna: sen pitää antaa tiedot onko verkossa Eulerin ketjuja, avoimia vai suljettuja, ja onko Eulerin verkko. (ks. Määritelmä 12.7.1 ja Lause 12.7.2).

10. Hamiltonin verkko

Kokeile Hamiltonin verkon triviaalialgoritmin animointia. Animointifunktio on `AnimoiHK.m`. Koeta ensin pienempiä verkkoja, esimerkiksi tehtävän 4 verkko, ja lopuksi vaikkapa (www-sivulta tai lataamalla `load ham1.txt`)

```
0 1 0 1 0 1 0 1 1 1
1 1 1 1 1 1 0 0 0 1
0 1 1 1 0 1 1 0 1 1
1 1 1 1 0 1 0 0 1 1
0 1 0 0 0 1 0 0 0 1
1 1 1 1 1 0 1 1 1 1
0 0 1 0 0 1 1 0 1 0
1 0 0 0 0 1 0 1 0 0
1 0 1 1 0 1 1 0 0 1
1 1 1 1 1 1 0 0 1 1
```

Eräs suljettu Hamiltonin ketju viimeisessä on: _____