Inlämningsuppgift
MMG200–Linjär algebra

HT07

Uppgift 1 - Grafer och flygplansrutter

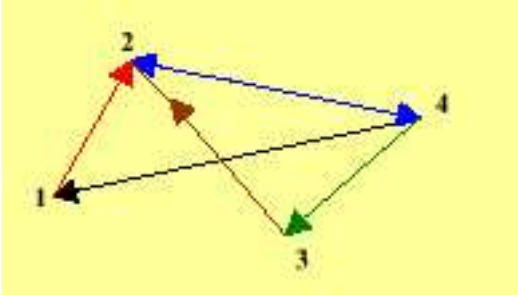
En så kallad *riktad graf* består av ett antal numrerade *noder* (punkter) sammanbundna med *riktade länkar* (pilar). I exemplet i figur 1 har vi 4 noder och 6 riktade länkar.

Kopplingsmatrisen till en riktad graf definieras som matrisen A med element $a_{ij} = 1$ om det finns en riktad länk från nod nr i till nod nr j, $a_{ij} = 0$ annars. Kopplingsmatrisen till den riktade grafen i figur 1 blir alltså

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \end{bmatrix}.$$

Flygplansrutterna hos ett flygbolag kan beskrivas med en riktad graf där flygplatserna är noder och direktrutterna är riktade länkar. Anta att ett flygbolag i Kalifornien har följande enkla flygningar:

från	till
San Francisco	Fresno
San Francisco	Monterey
Los Angeles	San Francicso
Los Angeles	Sacramento
Sacramento	San Francisco
Sacramento	Fresno
Fresno	Sacramento
Fresno	Los Angeles
Monterey	Los Angeles



Figur 1: En riktad graf

Uppgifter

a) Numrera städerna enligt: (1) Los Angeles, (2) San Francisco, (3) Monterey, (4) Fresno, (5) Sacramento. Rita motsvarande riktade graf.

b) Ta fram kopplingsmatrisen och lägg in den i MATLAB.

c) Beräkna A^2. Elementet (i, j) i A^2 är antalet rutter från i till j som använder exakt två enkla flygningar dvs med exakt en mellanlandning. Förklara detta genom att i detalj studera hur elementet $(2, 1)$ i A^2 kan tänkas räknas ut.

d) Beräkna $I + A + A^2$. Vad säger elementet (i, j) i denna matris om flygrutterna?

e) Vilket är det största antal flygningar som behövs mellan två städer? Räkna ut och förklara med matrisalgebra.

f) Bland andra är flygningen Sacramento - San Francisco kritisk för flygbolaget. I vilket avseende? Vad händer om man lägger ner den? Förklara med matrisalgebra!

g) Kan du addera endast en enkel flygning så att man kan flyga mellan vilka två städer som helst med högst en mellanlandning? Om inte, addera ytterligare flygningar (men så få som möjligt) så att detta går. Använd matrisalgebra och förklara hur du gör!

Inlämning: Svar på frågorna med tillhörande MATLAB-beräkningar.

Uppgift 2 - Kryptografi

Vi ska se på ett enkelt sätt att koda och avkoda hemliga meddelanden med hjälp av matristransformationer samt även hur man kan avslöja kodnyckeln dvs knäcka koden. Det okodade meddelandet kallar vi *klartext* och det kodade kallar vi *chiffer*.

En enkel kodnyckel får man om man ersätter varje bokstav i alfabetet med en annan, ett så kallat substitutionschiffer. Det är enkelt att knäcka en sådan kod eftersom man vet frekvenserna av olika bokstäver i ett språk.

Vi gör nu så att vi grupperar bokstäverna i klartexten och kodar grupp för grupp i stället för bokstav för bokstav. Om antalet bokstäver i en grupp är n kan vi ersätta dessa med en grupp av n chiffertecken genom en linjär transformation, dvs genom multiplikation med en nxn- matris. Vi betraktar alltså en grupp av n tecken som en kolonnvektor. En förutsättning är naturligtvis att vi kan ersätta tecken med tal men det hjälper MATLAB-funktionen `abs` oss med. `abs` konverterar från tecken till heltal och om vi använder en heltalsmatris som kodnyckel så blir motsvarande chiffer heltal, som sedan konverteras till tecken med MATLAB-funktionen `char`.

Så här kan en kodningsfunktion i MATLAB se ut för fallet $n = 4$, A är en given kodningsmatris av ordning 4×4:

```
function f=encode(p,A)
p=[p,'   '];
m=floor(length(p)/4);
pp=abs(reshape(p(1:4*m),4,m));
cc=A*pp;
f=char(reshape(cc,1,4*m));
```

Avkodning innebär den ”inversa” proceduren dvs lösning av linjära ekvationssystem. För att kunna läsa meddelandet krävs att man känner till nyckeln - matrisen A - som vi förutsätter är icke-singulär.

Uppgifter

a) Låt $n = 4$. Skriv en funktionsfil i MATLAB för avkodning, gärna med inspiration från kodnings-funktionen ovan. Hur mycket behöver Du egentligen ändra i den för att det ska bli avkodning i stället? Tolka meddelandet i variabeln *code_1* i filen */users/mdstud/hasse/krypto.mat*, som Du alltså kan ta in till MATLAB med kommandot `load` efter det att du kopierat den till eget directory. Du kan också hämta filen *krypto.mat* från kurshemsidan.

Den använda matrisen är:

$$A = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 2 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

Anmärkning: Om Du kodar själv: Begränsa Dig till små heltal i matrisen.

b) Om man känner till de n^2 första tecknen i klartexten (och motsvarande chiffer förstås) så kan man avslöja nyckeln (matrisen). (Man bör alltså undvika traditionella hövliga inledningsfraser i dessa sammanhang.) Vi antar att dessa inledande tecken är sådana att motsvarande vektorer blir linjärt oberoende. Man kan då bestämma matrisen A genom att lösa ett system med flera högerled dvs även detta med `slash` eller `backslash` (om man transformerar) i MATLAB. Låt $n = 4$. Skriv en funktionsfil i MATLAB som knäcker koden dvs bestämmer A, och använd den för att att läsa meddelandet i variabeln *code_2* i *krypto.mat*, som inleds med

The significant problems

Varning: Se upp med avrundningsfel. Matrisen ska vara en heltalsmatris.

c) För meddelandet

Nej Nej - vektorerna ej oberoende ...

som finns i variabeln *code_3* (i samma fil som ovan), blir vektorerna linjärt beroende då $n = 4$. Även detta kan relativt enkelt klaras av, med en liten modifikation av m-filen kanske, under förutsättning att vi känner till mera av klartexten, i detta fall 20 tecken . Om Du inte redan har skapat din fil med tanke på detta, så fixa till den så att Du kan avslöja A och läsa hela texten.

Ledning: Överbestämda ekvationssystem.

Inlämning: Funktionsfilerna och körningsresultat dvs klartexterna och nycklarna (matriserna).