Stærðfræði og reiknifræði – Skiladæmi 12

Við erum aftur með tvískipt verkefni og í þetta skipti eru það tveir hlutar sem leysa skal með blaði og blýanti, A og B. Skannið blaðið/blöðin þegar þið eruð búin t.d. með Camscanner og skilið í Canvas. Leysið svo C-hluta í Jupyter og skilið PDF-skjali sem búið er til í vafra undir S12C í Canvas. Skilafrestur er til kl. 22:30 á þriðjudag í næstu viku.

Dæmin eru úr kafla 5.3 (https://cs.hi.is/strei/kafli05.html#fylki-og-net) í fyrirlestrarnótum, og myndskeið með fyrirlestri úr kaflanum er undir upptökur í Canvas. Í þessum kafla eru 3 æfingadæmi sem gert er ráð fyrir að séu leyst á blaði og þau gegna hlutverki undirbúningsdæma fyrir A og B hluta. Lausnir á þeim verða birtar á föstudaginn.

A. Grannafylki

Stefnt net hefur leggi (eða örvar) frá hnút 1 til 3 og 4, frá 2 til 1 og 3, frá 3 í alla hina og frá 4 til 1.

1. a) Teiknið netið, þannig að hnútarnir séu í hornum fernings svona:
   $\qquad\begin{matrix} 1 & 2\\4&3\end{matrix}$
   b) Finnið grannafylkið A

1. Reiknið A^2 og svo $A^4 = (A^2)^2$ (sýnið útreikninga). Hvaða upplýsingar gefur (i,j)-stak A^4?

1. Lát $F(k)$ tákna fjölda mismunandi vega af lengd $\leq k$ sem tengja annað af efri hornum ferningsins í lið 1a við annað af neðri hornunum. Útskýrið hvernig hægt væri að reikna $F(k)$ (þið getið svarað með reikniriti eða í orðum).

B. Flæðinet

1. Teiknið stefnt net með 6 hnúta og örvar nr. 1–5 frá hnút k til hnúts $k+1$, $k=1,\ldots 5$, og auk þess örvar nr. 6–9 frá hnút 1 til 3, 2 til 4, 3 til 5 og 4 til 6 (alls 9 örvar). Búið til legufylki fyrir netið.

1. Látið flæðið í netinu í lið 1 vera $x_j = 2$ fyrir örvar $j=1,\ldots,5$, og $x_j = 3$ fyrir örvar $j=6,\ldots,9$. Notið varðveislujöfnu flæðis (sjá kafla 5.3.6), $Ax + s = 0$, til að ákvarða hvaða hnútar verða lindir og hvaða hnútar verða ósar?

1. Notum nú sama net og í lið 1 en annað flæði en í lið 2: Hnútur 1 er lind með innflæði 10 og hnútur 6 er ós með útflæði 10. Auk þess er gefið að flæðið í ör j er $x_j = j,\; j = 6,\ldots,9$. Notið varðveislujöfnuna til að ákvarða flæðið í hinum örvunum.

C. Python og flæði

Í þessu dæmi notum við sama net og í dæmi B1, nema hvað við byrjum að telja í 0. Í þetta sinn notum við Python. Upplýsingar um netið og flæðið (sem er ekki það sama og áður) eru í skrá cs.hi.is/strei/flaedi.txt (http://cs.hi.is/strei/flaedi.txt). Dálkarnir `b` og `e` gefa byrjunar- og endahnút hverrar örvar, `x` gefur flæði eftir henni og `lengd` gefur lengd hennar. Hér eru nokkrar skipanir sem nota þarf (margar eru upprifjun):

```
f = urlopen("url")                  Opnar skrá sem er á netinu
np.loadtxt(f, int, skiprows=n)      Skilar int-fylki (sjálfgefið float)
(d1,d2,d3) = np.loadtxt(...).T      Skilar skrá með 3 dálkum í 3 vigrum
max(a,b)                            Sú stærri af tölunum a og b
x.max()                             Stærsta stak í numpy-vigri x
len(x)                              Lengd vigurs x
np.zeros((m,n), int)                m x n heiltölu-núllfylki
A @ x                               Fylki sinnum vigur
disp(x)                             Skrifar út x (disp er skilgreint í "BYRJA")
N = nx.DiGraph(G)                   Býr til stefnt net með grannafylki G
nx.draw(N, with_labels = True,      Teiknar N (það gæti þurft að keyra nokkrum
                node_color = 'y')           sinnum til að staðsetja hnúta vel).
D = dijkstra(L)                     Skilar D[i,j] = lengd stysta vegar frá hnút i
```

In [1]:
```
#BYRJA -- Keyrið til að frumstilla
import numpy as np, networkx as nx
from urllib.request import urlopen
from scipy.sparse.csgraph import dijkstra
# disp(x,y...) skrifar x,y... með 3 aukastöfum
def disp(*args): print(*(f'{a:.3f}' if isinstance(a,float) else a for a in args))
np.set_printoptions(precision=4, floatmode='fixed', suppress=True, linewidth=150)
```

1. Lesið skrána `flaedi.txt` inn í fimm vigra, `ör`, `b`, `e`, `x` og `lengd` og prentið út niðurstöðuna. Ákvarðið m = fjölda hnúta og n = fjölda örva með `max` og `len`

In [197]:
```
#C1
f = urlopen('http://cs.hi.is/strei/flaedi.txt')
(ör, b, e, x, lengd ) = np.loadtxt(f, int, skiprows=1).T

print("Ör:", ör)
print("b:", b)
print("e:", e)
print("x:", x)
print("lengd:", lengd)

m = max(e)+1
n = len(ör)

print("Fjöldi hnúta:", m)
print("Fjöldi örva:", n)
```

```
Ör: [0 1 2 3 4 5 6 7 8]
b: [0 1 2 3 4 0 1 2 3]
e: [1 2 3 4 5 2 3 4 5]
x: [50 40 30 30 90 50 10 60 60]
lengd: [200 400 350 800 100 500  50 300 600]
Fjöldi hnúta: 6
Fjöldi örva: 9
```

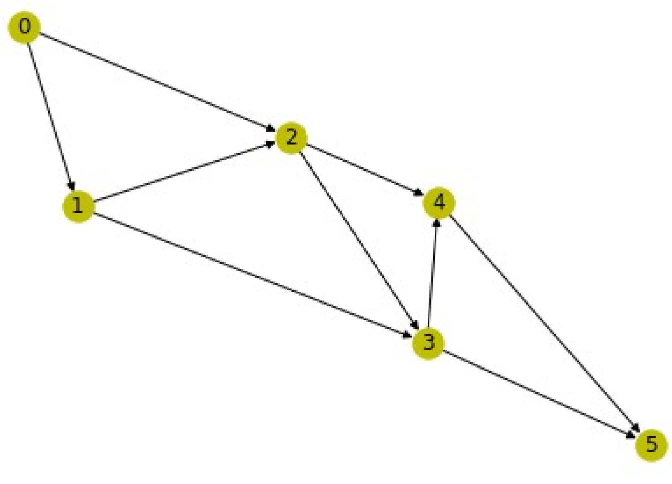
1. Búið til grannafylki, G, með því að byrja með $m\times m$ núll-fylki og lykkja svo með i yfir allar örvarnar og setja $g_{b_i e_i} = 1$ inni í lykkjunni (í Python `G[b[i],e[i]] = 1`). Teiknið í framhaldi netið með `networkx`

In [190]:
```
#C2
G = np.zeros((m,m), int)

for i in range(0, n):
    G[b[i],e[i]] = 1
print(G)

N = nx.DiGraph(G)
nx.draw(N, with_labels = True,  node_color = 'y')
```

```
[[0 1 1 0 0 0]
 [0 0 1 1 0 0]
 [0 0 0 1 1 0]
 [0 0 0 0 1 1]
 [0 0 0 0 0 1]
 [0 0 0 0 0 0]]
```



1. Búið líka til legufylki A með samskonar lykkju og í lið 2 (þið þurfið að setja -1 og 1 inn í fylkið á rétta staði). Finnið lindir og ósa og hve mikið flæðir inn og út með því að nota varðveislujöfnuna.

In [175]:
```
#C3
A = np.zeros((m,n), int)

for i in range(n):
    A[b[i], i] = -1
    A[e[i], i] = 1
print("A:")
print(A)

marg = A*x

S = [None]*6

for y in range(m):
    S[y]=sum(marg[y])*-1


print("S:", utk)
print("-------------")

for i in range(len(S)):
    if S[i] > 0:
        print(i, "er lind. Streymi:", S[i])
    if S[i] < 0:
        print(i, "er ós. Streymi:", S[i])
    if S[i] == 0:
        print(i, "er hvorki lind né ós. Streymi: 0")
```

```
A:
[[-1  0  0  0  0 -1  0  0  0]
 [ 1 -1  0  0  0  0 -1  0  0]
 [ 0  1 -1  0  0  1  0 -1  0]
 [ 0  0  1 -1  0  0  1  0 -1]
 [ 0  0  0  1 -1  0  0  1  0]
 [ 0  0  0  0  1  0  0  0  1]]
S: [100, 0, 0, 50, 0, -150]
-------------
0 er lind. Streymi: 100
1 er hvorki lind né ós. Streymi: 0
2 er hvorki lind né ós. Streymi: 0
3 er lind. Streymi: 50
4 er hvorki lind né ós. Streymi: 0
5 er ós. Streymi: -150
```

1. Hægt er að nota reiknirit sem kennt er við Dijkstra til að finna stystu fjarlægð milli tveggja hnúta í neti (Scipy útgáfu þess er lýst stuttlega að ofan). Ef við sendum grannafylki inn í Dijkstra er fjarlæðin reiknuð með því að telja örvar í stysta vegi milli hnútanna.

   a) Finnið lágmarksfjölda örva í vegi frá hnút 0 til 5.

   b) Hægt er að búa til lengdafylki L með eins lykkju og notuð var til að finna grannafylkið, nema við setjum l_{ij} = lengd örvarinnar $i\to j$. Gerið það, og finnið í framhaldi lengd stysta vegar frá 0 til 5.

In [196]:
```
#C4
print("a)")
L = np.zeros((m,m), int)

for i in range(n):
    L[b[i], e[i]] = lengd[i]


D = dijkstra(G)
print("Lágmarksfjöldi örva:", D[0,5])

print("b)")
D = dijkstra(L)
print("Lágmarksfjarlægð:", D[0,5])
```

```
a)
Lágmarksfjöldi örva: 3.0
b)
Lágmarksfjarlægð: 850.0
```

In [ ]: