Az alábbi irányított G gráf csúcsai a, b, c, d, e, f, élei pedig az alábbi éllistával adottak:

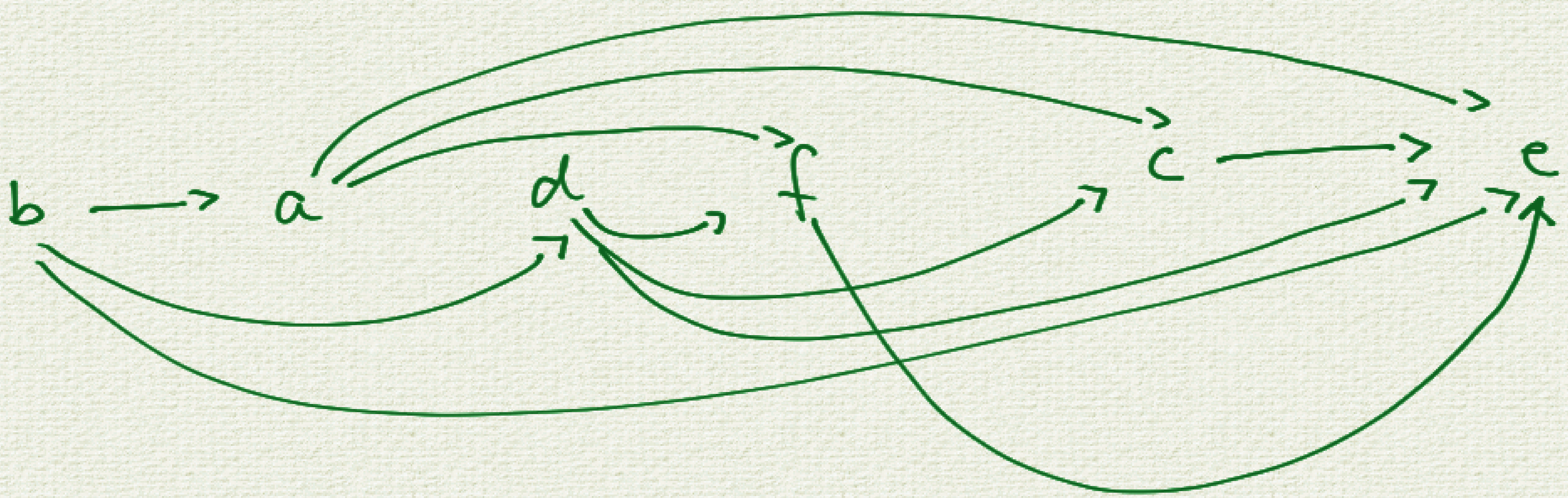
a $:c(8), f(6), e(12);$ **b** $:a(3), d(-8), e(-5);$ **c** $:e(-1);$ **d** $:c(2), e(4), f(-2), ;$ **e** $:-;$ **f** $:e(1);$

Topologikus sorrend-e ebben a gráfban a csúcsok b, a, d, f, c, e sorrendje és miért?

Az alábbi irányított G gráf csúcsai a, b, c, d, e, f, élei pedig az alábbi éllistával adottak:

a :$c(8), f(6), e(12)$; **b** :$a(3), d(-8), e(-5)$; **c** :$e(-1)$; **d** :$c(2), e(4), f(-2), ;$ **e** :$-$; **f** :$e(1)$;

Topologikus sorrend-e ebben a gráfban a csúcsok b, a, d, f, c, e sorrendje és miért?



Top sorrend : $\forall$ él $x \to y$

Adott egy 100 hosszú tömb, melyben a $100, 99, 98, \ldots, 3, 2, 1$ számok vannak ebben a csökkenő sorrendben. Melyik rendező algoritmus futtatása során történik a legkevesebb **csere** az alábbi lehetőségek közül és miért?
A lehetőségek: buborékrendezés, beszúrásos rendezés, kiválasztásos rendezés.

Adott egy 100 hosszú tömb, melyben a $100, 99, 98, \ldots, 3, 2, 1$ számok vannak ebben a csökkenő sorrendben. Melyik rendező algoritmus futtatása során történik a legkevesebb **csere** az alábbi lehetőségek közül és miért?
A lehetőségek: buborékrendezés, beszúrásos rendezés, kiválasztásos rendezés.

buborék

100, 99, 98, ..., 3, 2, 1

1. körben: 99 csere

99, 98, ..., 3, 2, 1 | 100

2. körben: 98 csere

⋮

$\boxed{99 + 98 + \ldots + 2 + 1 \text{ csere}}$

beszúrásos

100 | 99 | 98, ..., 3, 2, 1

1 csere

99, 100 | 98, — " —

2 csere

98, 99, 100 | 97, ...

3

⋮

2, 3, ..., 100 | 1

99

1, 2, ..., 100

1 + 2 + 3 + ... + 98 + 99 csere

kiválasztásos

100, ..., 2, $\boxed{1}$ $\boxed{1}$

1, 99, ..., 2, 100 $\boxed{2}$

1, 2 | 98, ..., $\boxed{3}$ 99, 100

⋮

50 csere

1, 2, ..., 50, 51, ..., 99, 100

=> kész

(de ha ezt nem vesszük észre

=> maximum 50 csere lehet

=> ≤ 100 csere van

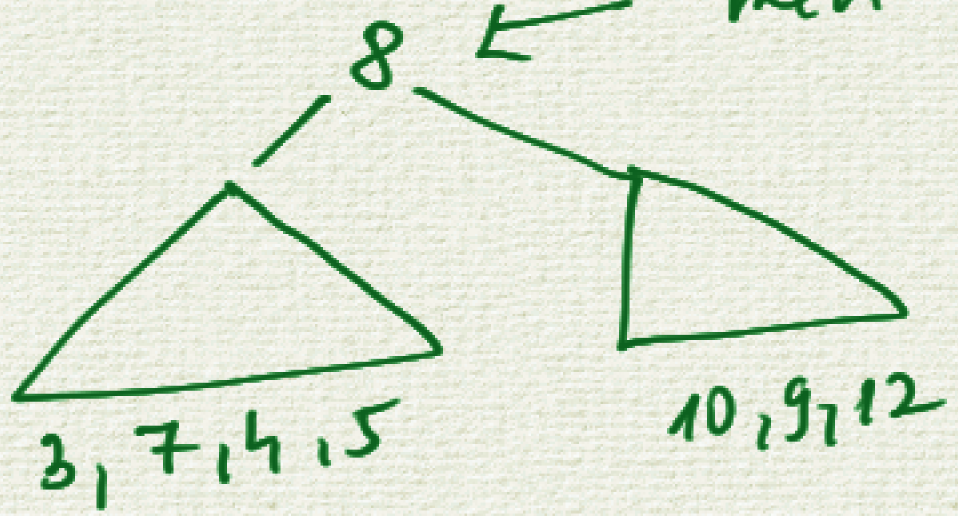
Egy bináris keresőfát preorder bejárással bejárva a számokat $8, 3, 7, 4, 5, 10, 9, 12$ sorrendben látjuk.
(a) Mi a fa gyökere és miért?
(b) Mely számok vannak a gyökér jobb- és bal-részfájában és miért?
(c) Hogy néz ki a fa és miért ez az egyetlen lehetőség?

Egy bináris keresőfát preorder bejárással bejárva a számokat 8, 3, 7, 4, 5, 10, 9, 12 sorrendben látjuk.
(a) Mi a fa gyökere és miért?
(b) Mely számok vannak a gyökér jobb- és bal-részfájában és miért?
(c) Hogy néz ki a fa és miért ez az egyetlen lehetőség?

pre(x): látogat(x)
pre(bal(x))
pre(jobb(x))

a)

b)



← mert preorder a gyökérrel kezd

← bin kerfa tul miatt

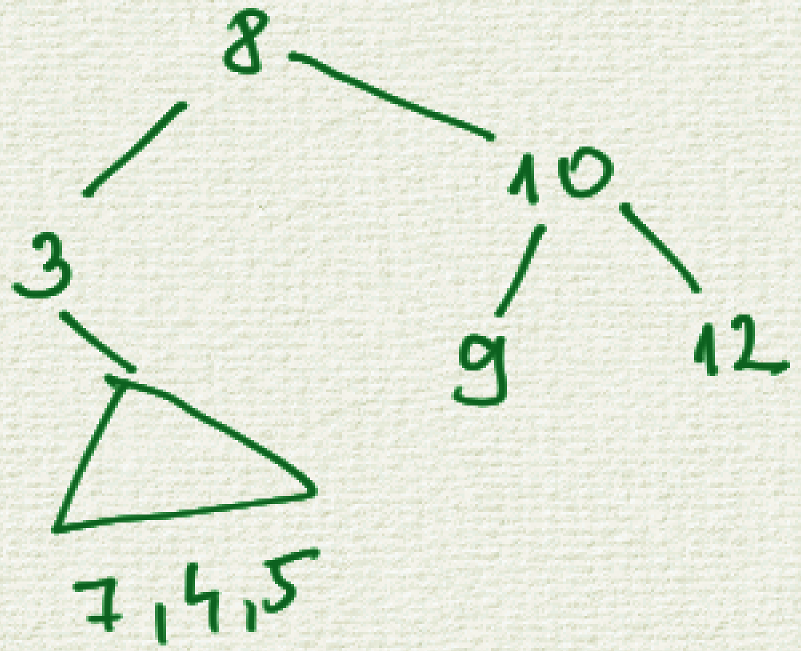
Sosincs választási lehetőség => egyértelmű a fa

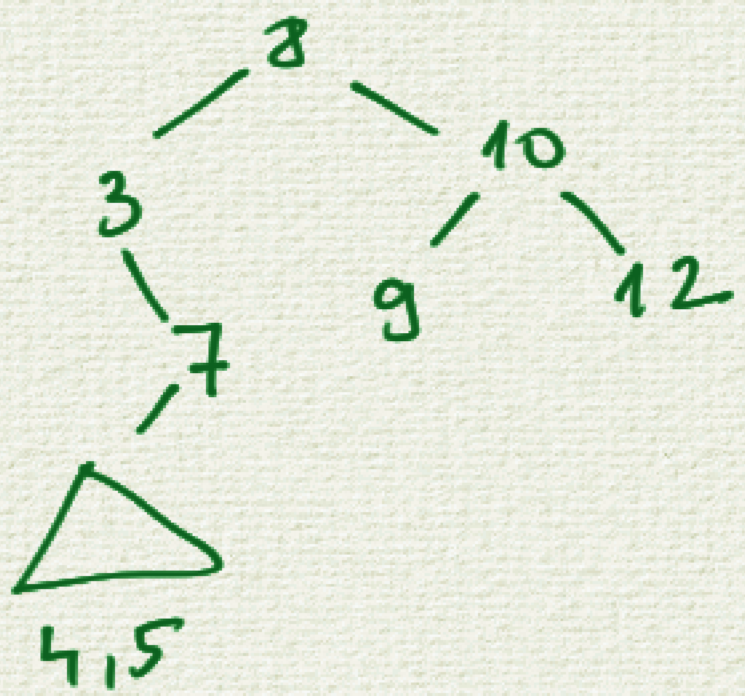
- gyökér jön a preorderből
- szétválás (bal/jobb) kerfa tul

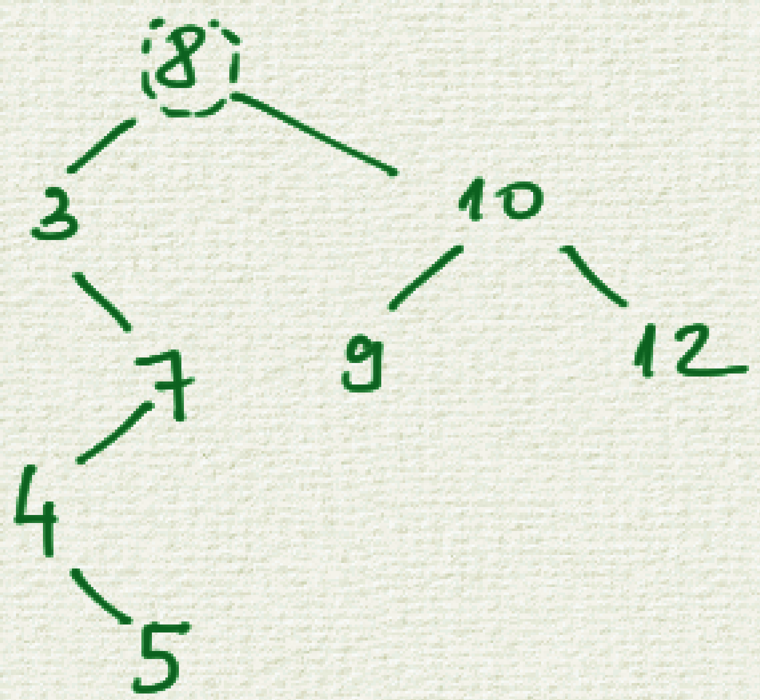
c)

preorder →

binkerfa →



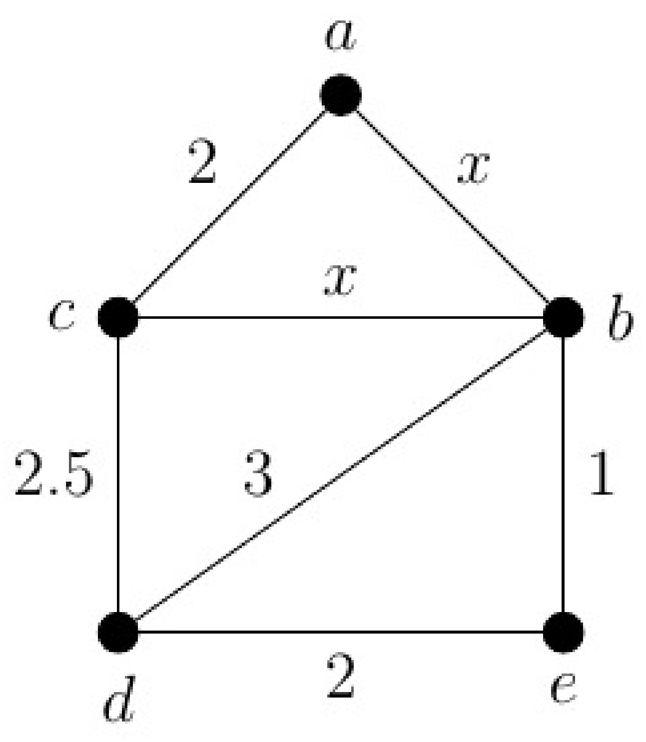




Az alábbi gráfon, ahol x értéke nem ismert, futtatjuk Prim algorimusát az a csúcsból egy minimális feszítőfa keresésére és azt tapasztaljuk, hogy az első három beválasztott él ab, be, ed ebben a sorrendben. (Az x értéke ugyanaz mindkét élen, ahol előfordul.)

(a) Mi lehet x értéke és miért?

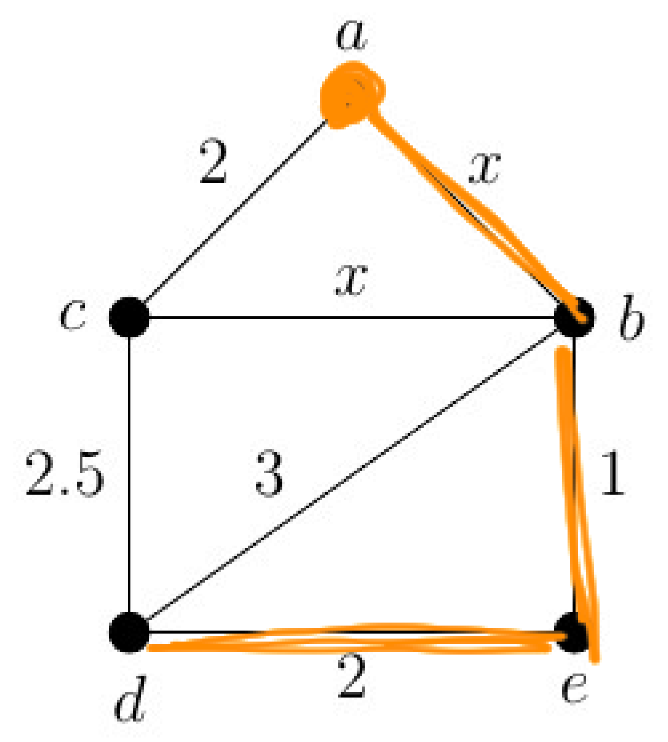
(b) Melyik élet (éleket) választhatja be az algoritmus a fenti három él után és miért?



Az alábbi gráfon, ahol x értéke nem ismert, futtatjuk Prim algorimusát az a csúcsból egy minimális feszítőfa keresésére és azt tapasztaljuk, hogy az első három beválasztott él ab, be, ed ebben a sorrendben. (Az x értéke ugyanaz mindkét élen, ahol előfordul.)

(a) Mi lehet x értéke és miért?

(b) Melyik élet (éleket) választhatja be az algoritmus a fenti három él után és miért?



a)

ab $\Rightarrow$ $\boxed{x \le 2}$ mert kben ac-t vette volna be

be $\Rightarrow$ $x \ge 1$ mert kben cb-t vette volna be

ed $\Rightarrow$ $\boxed{x \ge 2}$ mert kben bc-t veszi be és nem ed-t

$x = 2$

b) Nyitva van:

~~cd : 2,5~~

ac : 2

bc : 2 = x

ezekből az egyiket

~~bd : 3~~

Szomszédossági mátrixával adott egy n csúcsú irányítatlan G gráf. Adjon $O(n^3)$ lépésszámú algoritmust, ami G mátrixából előállítja annak az irányítatlan G_1 gráfnak a szomszédossági mátrixát, aminek csúcsai megegyeznek G csúcsaival és G_1-ben pontosan akkor van két csúcs között él, ha G-ben nincsen köztük él, de van közös szomszédjuk.

Szomszédossági mátrixával adott egy n csúcsú irányítatlan G gráf. Adjon $O(n^3)$ lépésszámú algoritmust, ami G mátrixából előállítja annak az irányítatlan G_1 gráfnak a szomszédossági mátrixát, aminek csúcsai megegyeznek G csúcsaival és G_1-ben pontosan akkor van két csúcs között él, ha G-ben nincsen köztük él, de van közös szomszédjuk.

G : A mátrixa $\rightsquigarrow$ G_1 gráf B mátrix

ötlet: $\forall$ i,j csúcspárra nézzük meg: • ~~van~~ nincs-e él i — j G-ben és , van-e közös szomszéd i — k — j => B[i,j] := 1

→ csinálok csupa-0 B-t (n×n-es)

pszeudokód:

```
ciklus i = 1-től n-ig:                          // végig az A mátrixon
    ciklus j = 1-től n-ig:
        ha A[i,j] == 0 :                        // nincs él i — j
            ciklus k = 1-től n-ig:
                ha A[i,k] == 1 és A[j,k] == 1 : // a k közös szomszéd
                    B[i,j] := 1                 // van él G_1-ben i — j
            c.v.
    c.v.
c.v.
```

n^2-szer fut le $O(n)$ ⇓ $O(n^3)$

$1 + O(n) \Rightarrow O(n)$

$n \cdot O(1) \Rightarrow O(n)$

$O(1)$

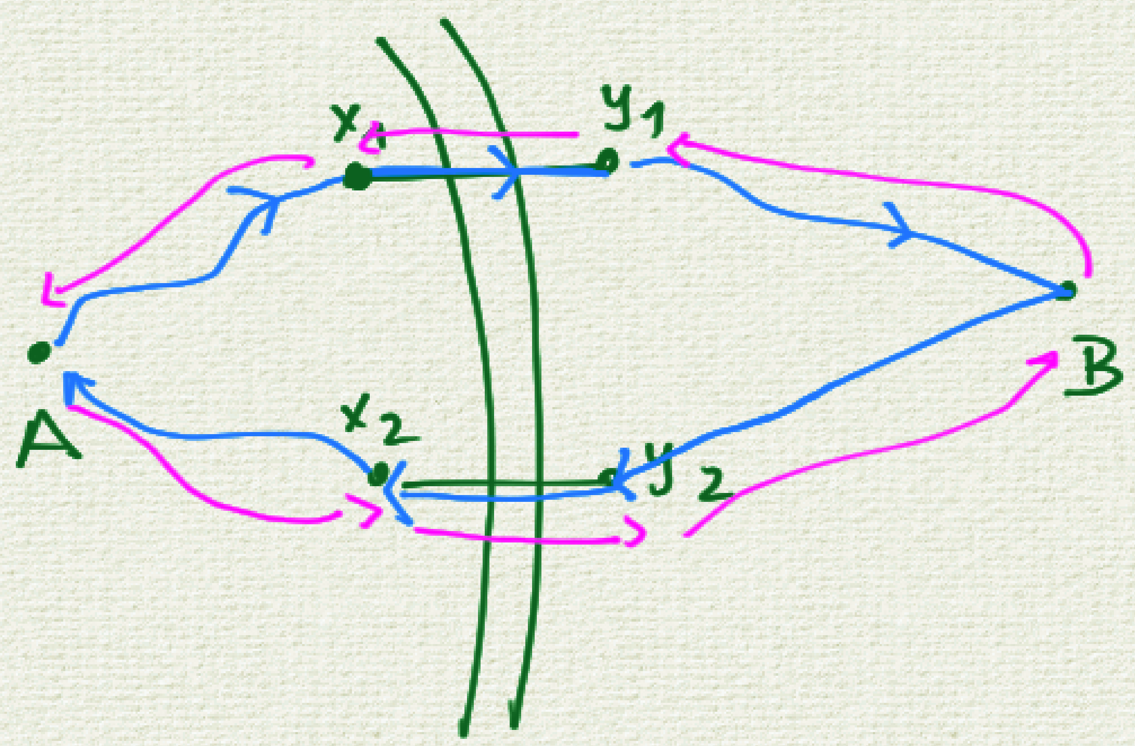
LSZ:

Szomszédossági mátrixával adott egy város úthálózatának élsúlyozott, irányítatlan gráfja: a csúcsok a csomópontok, az élek a csomópontok közötti közvetlen utak (melyek mindkét irányban használhatók, nincsenek egyirányú utcák), az élek súlya pedig azt mutatja, hogy mekkora adott útszakasz hossza. A várost egy folyó szeli ketté, a folyón két híd van, csak ezeken lehet átkelni a folyó egyik partjáról a másik partra (a hidak is egy-egy élet jelentenek a gráfban, azt ismerjük, hogy melyik élek a hidak).

Lakásunk az A csomópontban van, egy barátunk pedig a B csomópontban lakik, az A és B csomópontok a folyó ellentétes oldalán találhatóak. Barátunk karanténba került, de szeretnénk neki ajándékot vinni gyalog, a lehető legrövidebb utat megtéve, úgy, hogy az egyik hidat odafelé, a másik hidat visszafelé használjuk (azaz a lakásunkból indulunk, oda is érünk vissza és csak kétszer kelünk át a folyón, egyszer odafelé, egyszer visszafelé).
Melyik tanult algoritmust lehet alkalmazni, hogyan és miért, ha $O(n^2)$ lépésben meg akarjuk találni ezt a legjobb útvonalat (a szokásos módon n a csomópontok számát jelöli)?

Szomszédossági mátrixával adott egy város úthálózatának élsúlyozott, irányítatlan gráfja: a csúcsok a csomópontok, az élek a csomópontok közötti közvetlen utak (melyek mindkét irányban használhatók, nincsenek egyirányú utcák), az élek súlya pedig azt mutatja, hogy mekkora adott útszakasz hossza. A várost egy folyó szeli ketté, a folyón két híd van, csak ezeken lehet átkelni a folyó egyik partjáról a másik partra (a hidak is egy-egy élet jelentenek a gráfban, azt ismerjük, hogy melyik élek a hidak).

Lakásunk az A csomópontban van, egy barátunk pedig a B csomópontban lakik, az A és B csomópontok a folyó ellentétes oldalán találhatóak. Barátunk karanténba került, de szeretnénk neki ajándékot vinni gyalog, a lehető legrövidebb utat megtéve, úgy, hogy az egyik hidat odafelé, a másik hidat visszafelé használjuk (azaz a lakásunkból indulunk, oda is érünk vissza és csak kétszer kelünk át a folyón, egyszer odafelé, egyszer visszafelé).
Melyik tanult algoritmust lehet alkalmazni, hogyan és miért, ha $O(n^2)$ lépésben meg akarjuk találni ezt a legjobb útvonalat (a szokásos módon n a csomópontok számát jelöli)?



Ötlet
- Dijkstra
- em lehetőség van csak: legrövidebb út $A \sim x_1$ + $\boxed{y_1 - y_1 \text{ híd}}$ + legr. út y_1 és B között + legr. út y_2 és B között + $\boxed{x_2 - y_2 \text{ híd}}$ + legr. út $A \sim x_2$ között

Dijkstra A-ból

legr. út B-ből

- nem lehet itt híd => előbb ki kell dobni a hidakat => biztosítom ∀ híd 1×

jóság: + Dijkstra használható, mert ∀ élsúly ≥ 0

- magukat az utakat ⇐ Dijkstra algo-ból honnan tömbből jön

Algo
1) kidobom a 2 hidat a mátrixból → $O(1)$: ∞-re állítom a mátrixban
2) Dijkstra A-ból => legr. út A ~ x_1, ~ x_2
3) Dijkstra B-ből => legr. út B ~ y_1, ~ y_2

$\boxed{O(n^2)}$ $O(n^2)$

4) összeadok ∀ hosszat => legr. út lehetőség hossza) $O(1)$
5) legr. út ⇐ honnan tömbökből) $O(n)$

ÖSZ: $O(n^2)$