Nyílt címzéssel hasheltünk egy kezdetben üres, 11 elemű táblába nyolc kulcsot, a $h(k) = k$ maradéka 11-gyel osztva hash-függvényt és lineáris próbát használva. A beszúrások után az alábbi állapotot kaptuk:

(a) Magyarázza el, hogy hogyan került az 1-es kulcs a 10-es cellába a beszúrás során.
(b) A fenti táblából kitöröljük a 19-et, majd keressük a 30-at. Hogyan zajlik ez a keresés?
(c) A fenti táblából kitöröljük a 19-et és beszúrjuk a 20-at. Hogyan zajlik ez a beszúrás?

a) $h(1) = 1$

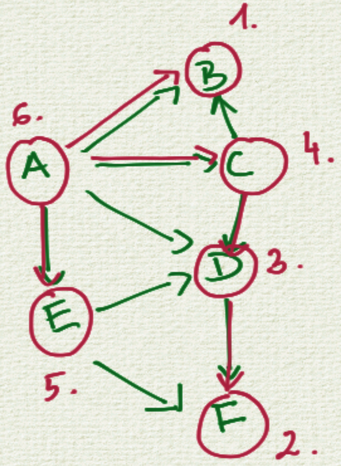
b) $h(30) = 22+8 = 8$

c) $h(20) = 11+9 = 9 \rightarrow$ 6-os cellába kerül

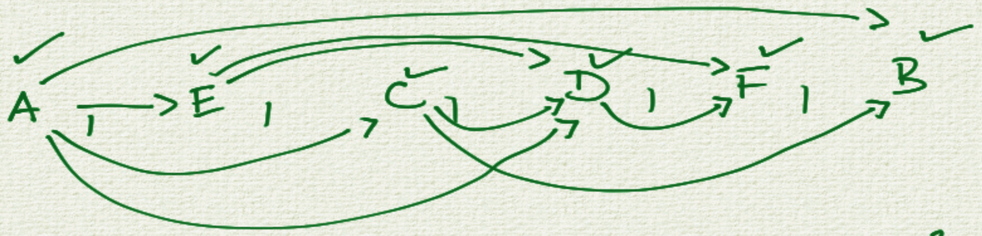
(a) Rajzolja fel az alábbi szomszédossági mátrixhoz tartozó irányított gráfot.

	A	B	C	D	E	F
A	0	1	1	1	1	0
B	0	0	0	0	0	0
C	0	1	0	1	0	0
D	0	0	0	0	0	1
E	0	0	0	1	0	1
F	0	0	0	0	0	0

(b) Hajtson végre egy mélységi bejárást ~~ezen a~~ bávhorran gráfon és a segítségével az órán tanult módszerrel döntse el, hogy ez a gráf DAG-e vagy sem.



befejezési számok fordított sorrendbe



→ topsorrend ⇒ G DAG

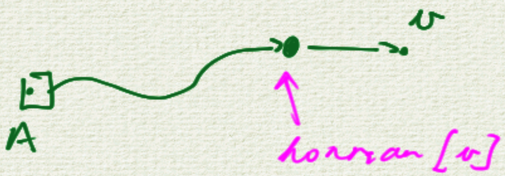
top sorrend ⇐⇒ G DAG

? Top-sorrend-e ?
azaz $\forall e'$ →
↓
azaz nincs e' ←

Dijkstra algoritmusát használjuk az A, B, C, D, E csúcsokból álló irányított, élsúlyozott G gráfban az A kezdőcsúcsból, eközben az *eddigi_legjobb* tömb így változik (az egyes sorok az *eddigi_legjobb* tömb változását mutatják egy-egy csúcs KÉSZ halmazba kerülése után).

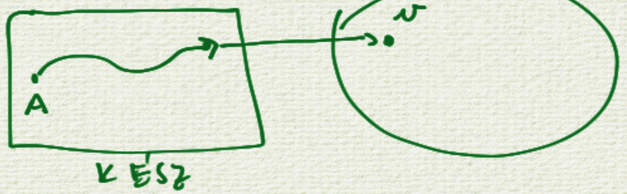
A	B	C	D	E
*	1	∞	5	∞
*	*	5	2	∞
*	*	4	*	∞
*	*	*	*	7
*	*	*	*	*

KÉSZ
A
B
D
C
E



Adja meg a *honnan* tömb értékét az egyes lépések után (egy ugyanilyen méretű táblázat formájában) és indokolja is meg a válaszát.

honnan gyűjtése
- ha $v \in$ KÉSZ : honnan jön a legrövidebb út (A-ból)
- ha $v \notin$ KÉSZ : honnan jön az a legrövidebb út, amíg KÉSZ-ben van-e'l

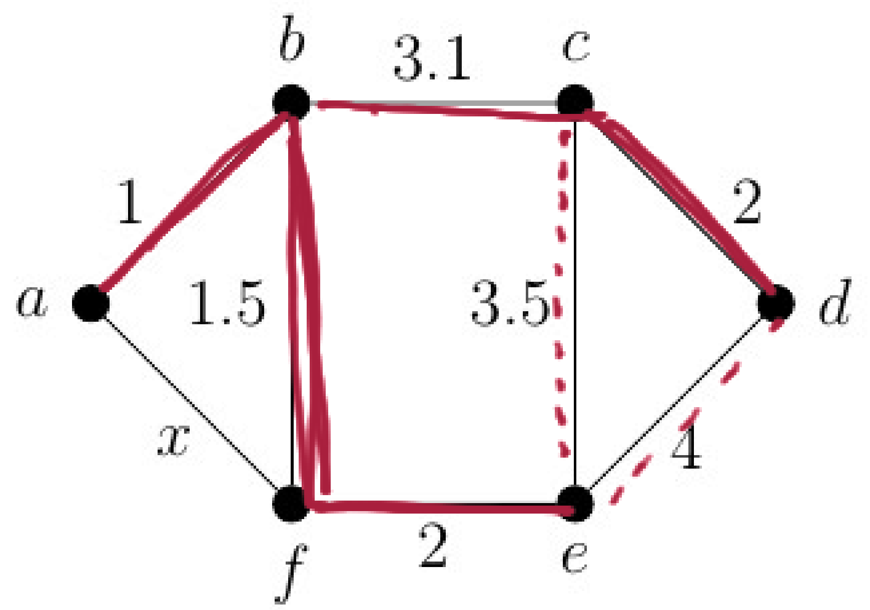


- ha eddigi-legjobb változik v-re ⇒ éppen KÉSZ-be került csúcsból jobb utat találok

honnan:

A	B	C	D	E
A	A	*	A	*
A	A	B	B	*
A	A	D	B	*
A	A	D	B	C
A	A	D	B	C

. Az alábbi gráfon, ahol az x élsúly nem ismert, futtatva Kruskal algorimusát egy minimális feszítőfa keresésére az első három él, amit az eljárás beválaszt: ab, bf, fe. Adja meg x összes lehetséges értékét (röviden indokolva válaszát) és írja le, hogy mely éleket és milyen sorrendben választja be ezután az algoritmus és miért.



Kruskal:

1) rendezi az éleket ↗

2) ∀ élre:

él + eddigiek kört csinálnak-e?

nem → beveszi

igen → ugorja

ab, bf ⇒ $x \geq 1.5$ mert különben vagy előre (ha $x < 1$) vagy másodikra ($x \geq 1$, $x < 1{,}5$) bevette volna

és itt bármi lehet, mert ezt sose választja, hiszen kört csinálna

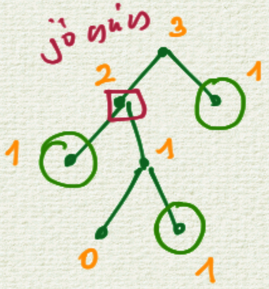
x = af valahol itt, de mindegy

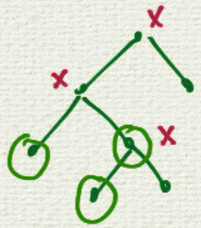
élek ab, bf, fe, cd, bc, ~~ce~~, ed

cd → beveszi; bc → beveszi; ce → kör lenne

5. Adott egy n csúcsú bináris keresőfa, amiben minden nem-levél csúcsnak két gyereke van és amiben néhány csúcs zöldre van színezve (a többi színtelen). Adjon $O(n)$ lépésszámú eljárást, ami eldönti, hogy van-e a fának olyan nem-levél csúcsa, aminek jobb és bal részfájában ugyanannyi zöld csúcs található.

LSZ





→ rossz fa

ötlet: 1) posztorder bejárás
"
jóság
| poszt(bal(x))
| poszt(jobb(x))
1 látogat(x) ← csinálunk vele valamit

2) feljegyzik, hogy x fiaiban (beleértve x-et is) hány zöld van → zöld[x] = ↑

3) mivel posztorder => x látogatásakor tudom, hogy a 2 részfában hány van

zöld[x]

O(n)

algo: posztorder bejárás és ha x-hez érsz (látogat x)

csúcsonként konstans munka plusszban

x = levél: ha x zöld => zöld[x] = 1, kben zöld[x] = 0

x ≠ nem-levél (=> 2 gyereke van => zöld[bal[x]], zöld[jobb[x]] már ki van számolva

ha zöld[bal[x]] == zöld[jobb[x]] => jó csúcs => kben zöld[x] := zöld[bal[x]] + zöld[jobb[x]] + 0/1 ← x zöld / x nem zöld

ha végig érsz, azaz x = gyökér is nem volt jó csúcs (a gyökér se jó) => "Nincs"

6. Egy téli napon egy országban rengeteg helyen esett ónos eső, a közlekedést felügyelő szervezet az utakat kategóriákba sorolta: 1-es a biztonságos útszakasz, 2-es a kicsit veszélyes, 3-as pedig a járhatatlan. Az ország úthálózatának ónos eső utáni helyzetét egy élsúlyozott irányítatlan gráf írja le, ahol csúcsok a városok, élek a városok között vezető utak, az élek súlya pedig a felügyelet által az útra kiadott kategória száma (1-es, 2-es vagy 3-as).

   El szeretnénk dönteni, hogy el tudunk-e menni az A városban levő lakásunkból a B városban levő nyaralónkba úgy, hogy végig biztonságos utakon haladunk. Melyik tanult algoritmust lehet alkalmazni, hogyan és miért, ha $O(n^2)$ lépésben választ akarunk kapni (szokás szerint n a csomópontok száma)?

∞ : nincs út
1 : biztonságos út
2 : kicsit veszélyes
3 : nagyon -"-

0,1 mátrix :
1 → jó út
0 → nincs út

• BFS/DFS

Algo: módosítsuk a mátrixot : akol. 2,3 → ∞ / 0
∞ → 0

jóság : új mátrixban már csak a használható utak

$O(n^2)$. BFS / DFS jó? A-ból
$O(1)$. bejárva[B] == 1 → elérhető, ha 0, ab akkor nem

LSZ: $O(n^2) + O(n^2) + O(1) = O(n^2)$