1. A szélességi keresés algoritmus bejárása során egy csúcs NEM kerülhet bele többször a sorba.
2. Annak eldöntése, hogy egy u csúcs szomszédos-e a v csúccsal szomszédossági éllistás ábrázolás esetén O(n), csúcsmátrixos reprezentáció esetén Theta(n^2)
3. Egy tetszőleges AVL fa inorder bejárása monoton növekvő sorozatot ad.
4. Egy n elemű AVL fában egy elem keresésének műveletigénye O(log n)
5. Tekintsünk egy 6-od fokú B+ fát. Egy levélben NEM lehet 6 adatrekordra hivatkozó mutató.
6. Egy tetszőleges AVL fában NEM mindig vannak azonos szinten a levelek.
7. Egy d-ed fokú B+ fában NEM csak beszúrással lehet bővíteni a belső csúcsokat.
8. Huffman kódolás során a kódolt üzenet NEM darabolható fel könnyen kisebb részekre, és NEM kódolhatóak külön-külön.
9. A szélességi keresés végén lehet olyan csúcs melynek távolsága végtelen (d = ∞).
10. B+ fában NEM szigorú monoton növekvő a sorrend (levelek).
11. Egy ***bináris fa*** (*p) csúcsára NEM IGAZ, hogy kiegyensúlyozott ha (p→b) egyensúlyára: p→b < 2, de ***AVL fa*** esetében IGAZ!
12. NEM csak az összefüggő gráfoknak lehet tranzitív lezártja.
13. Egy n csúcsú **sűrű** gráf esetében a sor alapú *Bellman Ford (BF)* algoritmus futtatása NEM hatékonyabb a
Floyd Warshall (FW) algoritmusánál (BF – O(n^4), míg a FW – O(n^3)), ritka gráf esetén ugyanaz a hatékonyság.
14. A Floy Warshall (FW) algoritmus (negatív összsúlyú kört nem tartalmaz) az inicializáláson felül összesen: n+n^2+n^3 iterációval határozza meg a legrövidebb utakat minden csúcspárra.
15. Dijkstra / Kruskal / Prim algoritmus mohó stratégiát követ.
16. Egy 20 hosszú szövegen és 7 hosszú mintán a Quick Search algoritmus NEM alkalmaz legalább 3 összehasonlítást.
17. NEM IGAZ az, hogy egy tetszőleges, irányított, élsúlyozott gráf és adott start csúcs esetén előállítható a legrövidebb utak fája.
18. A Bellman Ford algoritmus NEM használ minimum prioritásos sort a csúcsok kiterjesztéséhez.
19. A mintaillesztésben a next függvény egyik tulajdonsága, hogy: next(j+1) **≤** next(j)+1 (j∈1..m-1)
    A másik tulajdonsága: next(j) ∈ 0..(j−1)
20. A Kruskal algoritmusnak NEM kell minden esetben végig néznie a gráf minden csúcsát, hogy meghatározza a minimális feszítőfát.
21. Egy n csúcsú összefüggő, élsúlyozott, irányítatlan gráf minimális feszítőfája n-1 élből áll.
22. A Dijkstra algoritmus NEM képes detektálni negatív összsúlyú kört.
23. Egy tetszőleges, irányítatlan, összefüggő gráfnak több minimális feszítőfája is lehet.
24. Igaz az a tulajdonság, hogy: P[1..h+1] ⊐ P[1..j+1] <=> P[1..h] ⊐ P[1..j] ∧ P[1..h+1] = P[1..j+1]

A. Tetszőleges d-ed fokú (d>3) nem üres B+ fára:
    - ✓ Levelei azonos szinten vannak
    - ✓ levele lehet a nulladik szinten is
    - ✓ leveleinek kulcsai balról jobbra szigorú monoton nőnek.
    - ✓ levelei NEM tartalmazhatnak csak 1 kulcsot, ha a szintje legalább 1.

B. Egy tetszőleges AVL fában
    - ✓ a legnagyobb kulcsú elemének lehet bal gyereke
    - ✓ (*p) csúcs ($p \rightarrow b$) egyensúlyára: $p \rightarrow b < 2$
    - ✓ minden csúcs értéke különböző

1. Ha egy G irányított gráfnak van generátor csúcsa, akkor a gráf összefüggő.
2. A G = (V,E) gráfnak nem biztos, hogy részgráfja a G' = (V',E'), ha $V \in V'$ és $E \in E'$, mindkét irányított, vagy mindkettő irányítatlan.
3. NEM igaz az, hogy B+ fánál törléskor a törlendő kulcsot minden szintről törölni kell.
4. A Huffman kóddal tömörített szöveg kicsomagolása NEM kódtáblázat alapján történik.
5. A karakterenként kódoló tömörítések között a Huffman-kód minimális.
6. A szomszédossági éllistás ábrázolás NEM mindig helytakarékosabb a csúcsmátrixosnál.
7. A Prim algoritmus eredménye függhet attól, hogy melyik csúcsból indítjuk az algoritmust!
8. **<u>Ritka</u>** gráfon a Dijkstra algoritmus hatékonyabb mint a Floyd Warshall (FW). (Dijkstra – (n^2 log n), FW - O(n^3))
9. A Dijkstra algoritmust sűrű gráfon NEM érdemes a prioritásos sort kupaccal megvalósítani.
10. Dijkstránál csak a nem negatív élköltségek mellett garantált a helyes eredmény.
11. A mélységi gráfkeresés legjobb és legrosszabb esetben is Theta(n+m)
12. Az (u,v) visszaél detektálásakor a v befejezési ideje még nem definiált.
13. Minden kör felderíthető egy irányított gráfban mélységi gráfkereséssel.

a) Huffman kódolás során a tömörített szöveget karakterenként, változó hosszúságú bitsorozattal kódoljuk.
b) A karakterenként kódoló tömörítések között a Huffman-kód minimális.
c) A magassága NEM Theta(log n)

a) Egy B+ fában lehet olyan hasítókulcs, ami levélben nem szerepel.
b) Egy 6-odfokú esetén a B+ fában nem lehet a belsőcsúcsnak csak 2 gyereke (min. d/2 gyereke kell hogy legyen)

a) Ha egy G irányított gráfnak van generátor csúcsa, akkor a gráf összefüggő.
b) G = (V,E) irányítatlan gráfot csúcsmátrix-al ábrázolva a null elemek száma: n^2-2m.
c) G = (V,E) irányítatlan gráfot csúcsmátrix-al ábrázolva a null elemek száma: n^2-2m (ahol feltesszük, hogy nincsenek hurokélek).

a) Többféle szélességi fát kaphatunk, ha a szélességi keresés során egy csúcs szomszédjainak sorrendje tetszőleges.

b) Ha a feldolgozott él szürke csúcsba mutat, NEM biztos hogy irányított kört kapunk.

c) Az AVL fa egyben egy bináris keresőfa is.

a) A Dijkstra algoritmus működése során ha egy csúcs kikerült a prioritásos sorból, akkor az adott csúcsba már biztosan a legrövidebb út vezet a start csúcsból.

b) Előfordulhat, hogy a d érték végtelen marad.

a) Egy összefüggő irányított gráf mélységi bejárása NEM eredményez feszítőfát.

b) Ha az (u,v) egy mélységi fa éle, akkor u = Pi(v)

c) Ha u és v ugyanabban a mélységi fában van, akkor (u,v) él lehet keresztél.

d) Egy n elemű fa mélységi bejárása NEM azonosít n-1 faélt, mivel erdő esetén kevesebb is lehet.

a) A Floyd Warshall NEM határozza meg tetszőleges élsúlyozott gráfra a legrövidebb utakat minden csúcsra.

b) A Floyd Warshall képes felismerni a negatív összsúlyú kört.

c) A Dijkstra algoritmus **ritka** gráfok esetén hatékonyabb, mint a Floyd Warshall.

a) A Bellman Ford műveletigénye legjobb esetben aszimptotikusan megegyezik a KMP műveletidejével (ha m<<n)

b) Az inshift műveletideje aszimptotikusan megegyezik az initnext műveletidejével (Theta(m)).