

Rezultate importante cu grafuri și arbori.

1. Numărul de grafuri neorientate cu n vârfuri este $2^{n(n-1)/2}$
2. Numărul de grafuri orientate cu n vârfuri este $4^{n(n-1)/2}$
3. Numărul de grafuri orientate complete cu n vârfuri este $3^{n(n-1)/2}$
4. Fie G un graf orientat sau neorientat cu n vârfuri și m muchii/arce. Numărul de grafuri parțiale ale lui G este 2^m .
5. Fie G un graf orientat sau neorientat cu n vârfuri și m muchii/arce. Numărul de subgrafuri ale lui G este $2^n - 1$.
6. Suma gradelor vârfurilor unui graf neorientat este egală cu dublul numărului de muchii din graf: $\sum_{i=1}^n d(x_i) = 2 \cdot m$.
7. Suma gradelor interioare nodurilor unui graf orientat este egală suma gradelor exterioare nodurilor acestuia și este egală cu numărul de arce din graf:

$$\sum_{i=1}^n d^-(x_i) = \sum_{i=1}^n d^+(x_i) = m.$$

8. Fie G un graf neorientat. O secvență descrescătoare de n numere naturale

$n - 1 \gg d_1 \gg d_2 \gg d_3 \dots \gg d_n \gg 0$ se numește secvență grafică, dacă există un graf neorientat cu gradele vârfurilor: $d_1, d_2, d_3 \dots, d_n$.

Algoritm de verificare a unei secvențe dacă este secvență grafică:

- La fiecare pas unim prin muchii curente vârful curent i cu fiecare dintre cele d_i vârfuri care urmează și decrementăm gradele acestora
- După fiecare pas reordonăm vectorul după grade
- Algoritmul se termină după $n-1$ pași sau am găsit un vârf i pe care nu-l mai putem uni cu următoarele d_i vârfuri, deoarece am obține grad negativ, caz în care putem concluziona că secvența nu este secvență grafică.

Exemplu: secvența (4, 2, 2, 2, 2, 1, 1) este o secvență grafică, dar secvența (5, 4, 2, 2, 1, 1, 0) nu este secvență grafică

9. Graf biconex este un graf conex cu proprietatea că eliminând oricare nod al acestuia, graful rămâne conex.
10. Numărul de cicluri într-un graf neorientat complet este $2^n - 1 - n - n * (n - 1) / 2$, deoarece $C_n^3 + C_n^4 + C_n^5 + C_n^6 + \dots + C_n^n = 2^n - C_n^0 - C_n^1 - C_n^2$. (Un ciclu poate avea 3 vîrfuri, 4 vîrfuri, ..., n vîrfuri).
11. Numărul maxim de muchii pe care le poate avea un graf neorientat cu p componente conexe este $\frac{(n-p)(n-p+1)}{2}$, unde p - nr. de componente conexe, n - nr. de vârfuri
12. Numărul de cicluri hamiltoniene diferite într-un graf neorientat complet cu n vârfuri este $\frac{(n-1)!}{2}$ și într-un graf orientat complet cu n noduri este $(n-1)!$. Aceste afirmații se bazează pe presupunerea că ciclurile care sunt aceleași cu excepția punctului de plecare nu sunt luate în calcul separat.
13. Pentru a determina numărul de drumuri de lungime k dintre două noduri i și j ale unui graf (orientat sau neorientat), se calculează matricea $X=A^k$, unde A este matricea de adiacență iar $X[i][j]$ reprezintă numărul de drumuri de lungime k dintre i și j .
14. Un graf nu admite cicluri dacă și numai dacă $m - n + p = 0$, unde p este numărul de componente conexe. (Bravo, Liviu!)
15. Numărul ciclomatic: Numărul maxim de cicluri independente pentru un graf este dat de numărul ciclomatic este $m - n + p$, unde p - numărul de componente conexe.
16. Într-un graf neorientat conex, există un număr de $m - n + 1$ cicluri elementare
17. Un graf bipartit se numește complet dacă fiecare vârf din mulțimea A este adiacent cu fiecare vârf din mulțimea B .

Observație: Dacă numărul de vârfuri din mulțimea A este p , iar numărul de vârfuri din mulțimea B este q , graful bipartit complet se notează K_{p-q} și conține $p \cdot q$ muchii.

18. Un graf G este **planar** dacă poate fi desenat în plan astfel încât muchiile să nu se intersecteze decât în nodurile grafului. O astfel de desenare se numește reprezentare planară a lui G . Orice reprezentare a unui graf planar împarte planul în regiuni, numite fețe ale grafului, întotdeauna existând o față nemărginită, numită față exterioară.

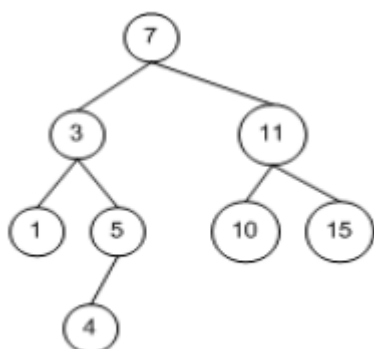
Formula lui Euler: Dacă G este un graf planar conex cu n noduri, m muchii și f fețe, atunci numărul fețelor este $f = m - n + 2$.

19. Se numește arbore un graf neorientat, conex și aciclic.

Următoarele afirmații sunt echivalente pentru un graf G cu n vârfuri și m muchii (i. $\Leftrightarrow$ ii. $\Leftrightarrow$ iii. $\Leftrightarrow$ iv. $\Leftrightarrow$ v. $\Leftrightarrow$ vi.)

- i. G este arbore
 - ii. G este graf aciclic cu $n-1$ muchii
 - iii. G este graf conex cu $n-1$ muchii
 - iv. G este maximal în raport cu proprietatea de aciclicitate (dacă se adaugă o singură muchie, devine ciclic)
 - v. G este minimal în raport cu proprietatea de conexitate (dacă se suprimă o singură muchie, devine neconex)
 - vi. Orice pereche de vârfuri este unită printr-un singur lanț
20. Orice arbore cu n noduri are **$n-1$** muchii.
21. Numărul arborilor cu n vârfuri este **n^{n-2}** (Teorema lui Cayley)
22. Fie numerele $d_1 \geq d_2 \geq \dots \geq d_n \geq 1$. Atunci există un arbore cu șirul gradelor $d_1, d_2, \dots, d_n$ dacă și numai dacă $\sum_{i=1}^n d(x_i) = 2n - 2$.
23. Se numește arbore binar un arbore cu proprietatea că fiecare nod are cel mult doi descendenți direcți (fii).
24. Se numește arbore binar strict un arbore cu proprietatea că fiecare nod (cu excepția frunzelor) are EXACT doi descendenți direcți (fii).
25. Un arbore binar complet este un arbore strict care are toate nodurile terminale situate pe același nivel
26. Un arbore binar complet cu n noduri terminale are $2 \cdot n - 1$ noduri.
27. Parcurgerea arborilor binari:
- i. Preordine (RSD)
 - ii. Inordine (SRD)
 - iii. Postordine (SDR)
28. Se numește arbore binar de căutare un arbore binar cu proprietatea că, pentru fiecare nod, cheia (informația) din succesorul său stâng este mai mică decât cheia sa, iar cheia (informația) din nodul său drept este mai mare decât cheia sa.

Un arbore binar de căutare este un arbore binar destinat îmbunătățirii timpului de căutare a informației.



OBSERVAȚIE: Eficiența prelucrării arborilor binari depinde de modul în care a fost creat arborele, putând fi $O(n)$ sau $O(\log_2 n)$. Dacă arborele are înălțimea maximă (fiecare nod are exact un descendent- arbore degenerat) complexitatea este $O(n)$ iar dacă arborele are înălțimea minimă (arbore complet sau aproape complet) atunci complexitatea este $O(\log_2 n)$.