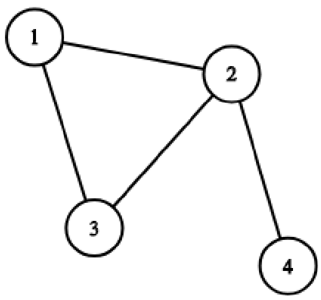
Grafuri și Arbori - noțiuni teoretice

Un graf este o rețea de **noduri** legate între ele prin **muchii**.

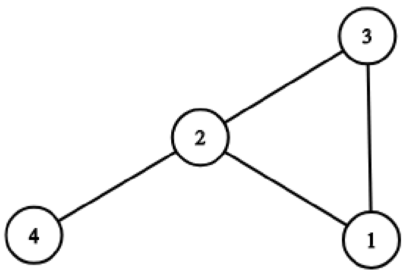
O definiție mai abstractă este următoarea: Un graf este o pereche formată din două mulțimi: O mulțime `X` de elemente, numite noduri și o mulțime `U` formată din perechi de elemente din `X`, numite muchii.

De regulă, elementele mulțimii nodurilor le notăm prin numere naturale începând cu `1`. Convenim ca în acest material să notăm cu `n` numărul de noduri și cu `m` numărul de muchii.

Dacă avem `X` = {`1`, `2`, `3`, `4`} și `U` = {(`1`, `2`), (`1`, `3`), (`2`, `3`), (`2`, `4`)}, putem spune că avem un graf format din trei noduri și patru muchii. O posibilă reprezentare grafică a sa este următoarea:

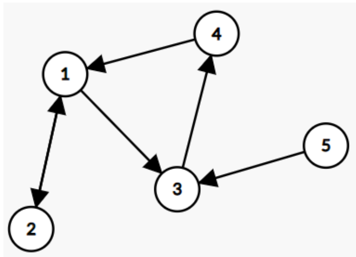


Subliniem din start că din punct de vedere al teoriei grafurilor, nu interesează poziția în plan (sau spațiu) a nodurilor ci numai modul în care sunt legate între ele. Astfel, graful din figura următoare este același cu cel din figura anterioară, pentru ambele mulțimile `X` și `U` fiind identice.



Graful de mai sus spunem că este unul **neorientat** întrucât pentru orice muchie (`a`, `b`) este indiferentă orientarea sa de la `a` la `b` sau de la `b` la `a`.

Altă categorie de grafuri o reprezintă cele **orientate**, în acest caz, fiecare muchie are bine stabilită orientarea. De obicei la aceste grafuri muchiile se desenează prin săgeți. Iată un exemplu de graf orientat:



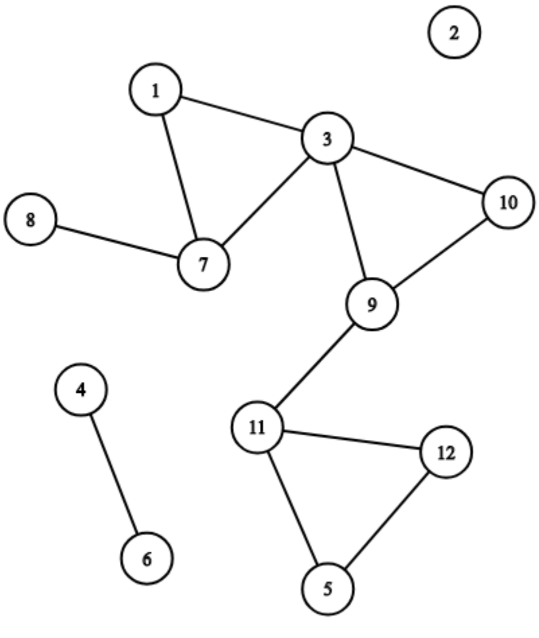
Pentru acest graf avem: X = {1, 2, 3, 4, 5} și U = {(4,1), (3,4), (5,3), (1,3), (1,2), (2,1)}.

Observăm că în acest caz este importantă ordinea în care scriem nodurile în cadrul unei muchii precum și faptul că toate muchiile sunt orientate (dacă avem orientare în ambele sensuri putem considera că avem două muchii). Multe dintre noțiuni precum și multe dintre rezultate sunt identice la cele două categorii de grafuri. Sunt și diferențe, pe care le vom puncta. De exemplu, se spune muchie la grafurile neorientate, pe când la cele orientate spunem arc. Se spune nod la grafurile neorientate și vârf la cele orientate. În acest material vom folosi noțiunile de nod și de muchie pentru ambele categorii.

În continuare vom studia detaliat grafurile neorientate iar la capitolul dedicat celor orientate vom puncta în principal deosebirile.

Grafuri neorientate

Fie graful din figura următoare, notat pe mai departe cu G_1.



G1

Vom prezenta anumite definiții și rezultate de bază pentru grafuri, folosindu-l pe acesta drept exemplu.

- X = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12}. Numărul său de noduri este n = 12.

- U = {(1,3), (3,10), (1,7), (3,7), (3,9), (9,10), (8,7), (4,5), (9,11), (11,12), (12,5), (5,11)}. Numărul său de muchii este m = 12.
- Pentru fiecare muchie putem face notația de mai sus sau notația inversă, fiind același lucru în cazul grafurilor neorientate. De exemplu, prima muchie din listă am fi putut-o nota și (3,1).
- Noțiunea de **adiacență** se folosește de regulă pentru noduri. Se spune că două noduri legate prin muchie sunt adiacente. În exemplul de mai sus 11 și 5 sunt noduri adiacente dar 9 și 5 nu sunt.
- Folosim noțiunea de **incidență** când spunem că un nod este extremitate a unei muchii. Astfel, nodul 9 este incident cu muchiile (3, 9), (10, 9) și (9, 11). Nodul 11 nu este incident cu muchia (5, 12).
- Numim **grad** al unui nod ca fiind numărul de muchii incidente cu acel nod. Astfel gradul nodului 11 este egal cu 3. Notația folosită este d(11) = 3. Notația d este împrumutată de la corespondentul cuvântului *grad* în limba franceză (*degre*). Nodurile cu grad 0 se mai numesc **noduri izolate**. În exemplul de mai sus izolat este nodul 2. Legat de grade avem și prima proprietate importantă:

$$d(1) + d(2) + ... + d(n) = 2\cdot m$$

Altfel spus, suma tuturor gradelor nodurilor unui graf neorienta este egală cu dublul numărului de muchii (deducem că suma gradelor este un număr par). Justificarea acestei formule este imediată: fiecare muchie contribuie cu 2 la suma totală, fiind incidentă cu două noduri, aducând 1 la gradul fiecăruia dintre ele.

- Numim **lanț** o secvență de noduri ce au proprietatea că oricare două vecine reprezintă capetele unei muchii a grafului. De exemplu 1, 3, 10, 9, 11 reprezintă un lanț în graful G_1, iar secvența formată din 1, 3, 9, 7, 8 nu reprezuintă lanț deoarece între 9 și 7 nu există muchie.
- Numim **lanț elementar** un lanț care are toate nodurile distincte.
- Numim **lanț simplu** un lanț care are toate muchiile distincte.

Exemple:

1, 3, 10, 9, 11	Lanț elementar, deci si simplu.	
8, 7, 3, 9, 10, 3, 1	Lanț neelementar dar simplu.	Lanțul nu este elementar pentru că se repetă nodul 3 dar este simplu pentru că nu este nicio muchie care să fie traversată de mai multe ori.
7, 3, 9, 11, 5, 12, 11, 9, 10	Lanț care nu este simplu (deci nu este nici elementar).	Observăm că munchia dintre 9 și 11 se repetă în cadrul lanțului.
1, 3, 9, 7, 8	Nu este lanț.	Între 9 și 7 nu avem muchie.

- Numim **ciclu** un lanț simplu, închis. Așadar, prin definiție, în cadrul unui ciclu nu putem avea o muchie care să se repete. Se pot însă repeta noduri, caz în care se poate împrumuta denumirea folosită la lanț și spunem că avem ciclu neelementar.

Exemple:

1, 3, 7, 1	Ciclu elementar.	Faptul că am scris nodul 1 de două ori nu înseamnă că acesta se repetă pe ciclu ci este doar un mod de a nota că lanțul se închide.
7, 1, 3, 7	Ciclu elementar.	Este vorba despre același ciclu, deci observăm că nu este importantă ordinea de scriere a nodurilor când îl notăm, ci structura ciclului.
1, 3, 10, 9, 3 7, 1	Ciclu neelementar.	Se repetă nodul 3.

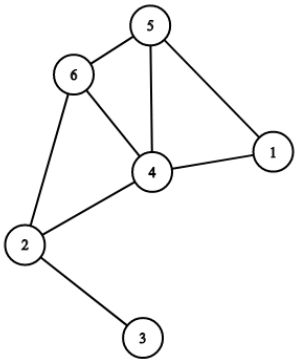
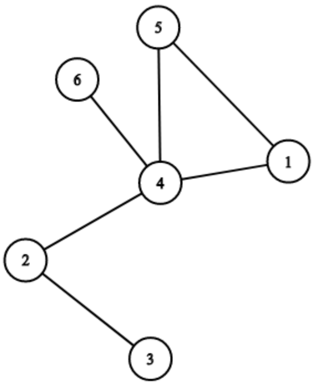
3, 9, 11, 5, 12, 11, 9, 10, 3	Nu este un ciclu.	Muchia 9, 11 se repetă și astfel nu este respectată definiția

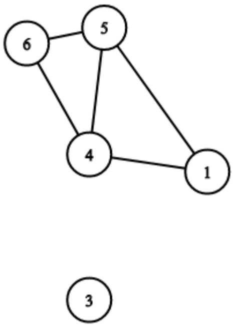
- Lungimea unui lanț și a unui ciclu se definește ca fiind numărul de muchii ale acestuia (atenție, nu numărul de noduri). Astfel, în exemplele anterioare, primul lanț are lungimea 4, al doilea are lungimea 6 și al treilea are lungimea 8. În cazul ciclurilor, primele două (fiind și identice) au lungimea 3 iar al treilea are lungimea 6.

Noțiunile de graf parțial și subgraf.

Numim **graf parțial** al unui graf dat, ceea ce rămâne din graful dat păstrând toate nodurile și eliminând eventual unele muchii, fără a adăuga muchii noi.

Numim **subgraf** al unui graf dat, ceea ce rămâne din graful dat eliminând unele noduri și doar muchiile incidente lor, deci nu și alte muchii și fără să adăugăm alte muchii.

G2		Graful dat.
G2’		În stânga avem un graf parțial al lui G_2 întrucât s-au păstrat toate nodurile și s-au eliminat muchiile (2,6) și (6, 5) fără să se fi adăugat muchii neexistente în graful inițial.

G2''		În stânga avem un subgraf al lui G_2 pentru că acesta s-a obținut eliminând din G_2 nodul 3 și numai muchiile incidente cu acesta, fără să se mai adauge sau să se elimine alte muchii. Subgraful s-a obținut prin eliminarea unui singur nod, dar se pot elimina și mai multe.

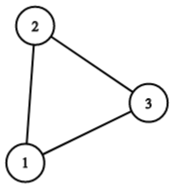
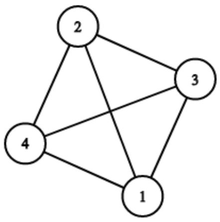
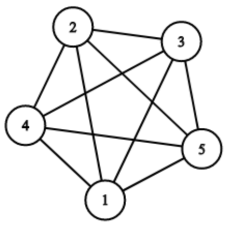
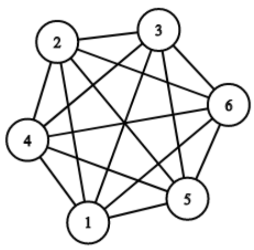
Pentru un graf dat, cu m muchii, numărul de grafuri parțiale ale sale este 2^m. Acest lucru se deduce ușor din definiție: practic orice submulțime a mulțimii muchiilor grafului dat induce un graf parțial al său. Pe de altă parte cunoaștem că o mulțime cu m elemente are 2^m submulțimi, și de aici obținem rezultatul de mai sus.

Graf complet, graf bipartit, graf regulat.

Prezentăm în continuare definiții și rezultate de bază pentru câteva categorii speciale de grafuri.

Graf complet

Un graf neorientat cu n noduri și cu muchie între oricare două noduri se numește **graf complet** de ordin n. În materialul de față de vom referi la el și cu notația K_n.

K_3	
K_4	
K_5	
K_6	

Numărul de muchii ale lui K_n este $n \cdot (n-1)/2$ sau, altfel spus combinări de n luate câte două, C_n^2.

Sunt multiple modalități de a justifica acest rezultat. Iată câteva:

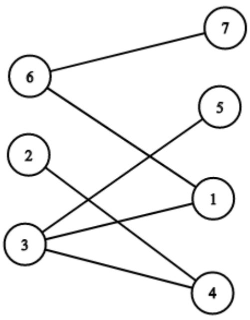
- Din fiecare nod se duc câte $n-1$ muchii (către toate celelalte noduri). Se obțin astfel $n \cdot (n-1)$ muchii. Însă orice muchie a fost numărată de două ori, dinspre fiecare dintre extremitățile sale.
- Pentru a nu număra de două ori muchiile, ne gândim că scriem extremitățile în ordine crescăroare. Astfel, din nodul 1 avem $n-1$ muchii către noduri cu număr de ordine mai mare. De la nodul 2 vom avea $n-2$ astfel de muchii (fără cea către 1), din nodul 3 avem $n-3$ muchii ... și așa mai departe, la final o singură muchie de la nodul $n-1$ la nodul n. Așadar avem $n-1 + n-2 + n-3 + ... + 1$. Acest număr reprezintă suma Gauss de $n-1$ adică $(n-1) \cdot n/2$.
- Ne putem gândi că fiecare muchie este o modalitate de a alege două elemente distincte dintr-o mulțime cu n elemente. De aici combinări de n luate câte două.

Combinând noțiunea de graf complet cu rezultatul prezentat anterior, anume că un graf cu m muchii are 2^m grafuri parțiale, și observând că orice graf cu n noduri este un graf parțial al lui K_n, obținem următorul rezultat, foarte important:

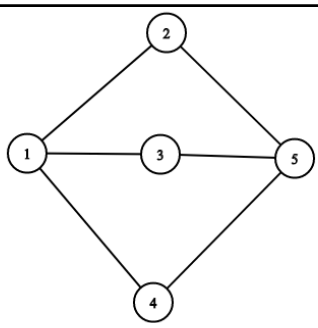
- Numărul de grafuri neorientate cu n noduri este $2^{C_n^2}$.

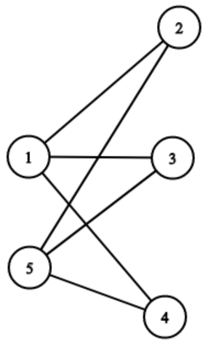
Graf bipartit

Un graf neorientat pentru care există posibilitatea de a-i partiționa mulțimea de noduri în două submulțimi nevide, așa încât orice muchie să aibă o extremitate într-o submulțime și cealaltă extremitate în cealaltă submulțime, se numește graf bipartit.

	Graful din figura alăturată este bipartit, iar submulțimile în care este partiționată mulțimea de noduri sunt {2, 3, 6} respectiv {1, 4, 5, 7}.

Modul de a realiza desenul, ca mai sus, este uzual când reprezentăm grafuri bipartite (cele două submulțimi ale partiției se scot în evidență scriindu-se fiecare pe câte o coloană). Totuși, nu este obligatoriu să facem astfel desenul. În tabelul de mai jos este un singur graf bipartit, desenat în două moduri.

	Figura de jos este obținută din cea de sus “trăgând” nodul 5 în stânga. Este vorba despre același graf întrucât nodurile și muchiile rămân aceleași (știm că două grafuri diferă doar prin componența mulțimilor de noduri și de muchii, nu prin modul de așezare al acestora într-un desen).

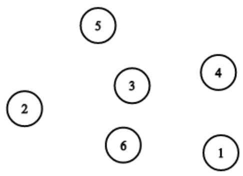
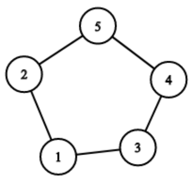
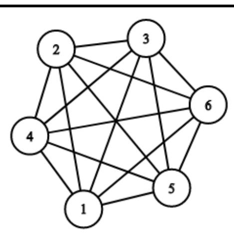
Următorul rezultat este foarte des utilizat în cazul grafurilor bipartite: un graf este bipartit dacă și numai dacă orice ciclu are lungime pară.

Observați că acest lucru se verifică pentru toate cele trei grafuri desenate mai sus. Poprietatea este ușor vizibilă la graful desenat primul în tabel.

Este întâlnită și noțiunea de graf **bipartit complet**. Să îl notăm $B_{s,d}$. Avem doi parametri întrucât la el trebuie să indicăm cardinalul la fiecare dintre cele două mulțimi. Astfel, $B_{s,d}$ este un graf cu s noduri în una dintre cele două mulțimi și d noduri în cealaltă și care **are muchie între firecare nod dintr-o mulțime și fiecare nod din cealaltă**. Astfel, $B_{s,d}$ are $s \cdot d$ muchii. Graful din figura de mai sus este un $B_{2,3}$.

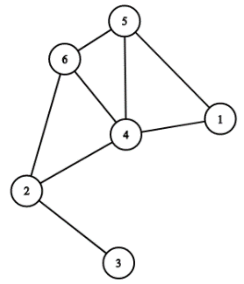
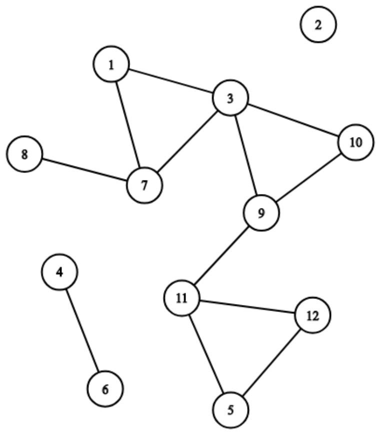
Graf regulat

Un graf în care toate nodurile au același grad se cheamă regulat. Când ne referim la grafuri regulate, în fața cuvântului "regulat" se mai scrie și valoarea comună a gradelor.

	`0-regulat` Graful fără muchii, deci cu toate nodurile de grad `0` este, evident, regulat.
	`2-regulat` Graful "poligon" este regulat
	`(n-1) - regulat` K_n este regulat

Conexitate

Un graf neorientat în care există lanț între oricare două noduri se numește graf conex.

	În figura alăturată este graful G_2 din exemplele prezentate anterior. Acesta este conex.
	Alături este graful G_1, folosit pentru exemplificare la începutul capitolului. Acest graf nu este conex (între nodurile 4 și 9, spre exemplu, nu este lanț).

Altfel spus, un graf este conex dacă se poate ajunge între oricare două noduri mergând doar pe muchii.

Apare și noțiunea de **componentă conexă**. Graful G_1 nu este conex, dar are trei componente conexe.

Prin definiție, **o componentă conexă este un subgraf conex și maximal cu această proprietate**. Termenul maximal se referă la faptul că respectivul subgraf nu mai poate fi extins cu noduri din graful inițial așa încât să se păstreze conexitatea.

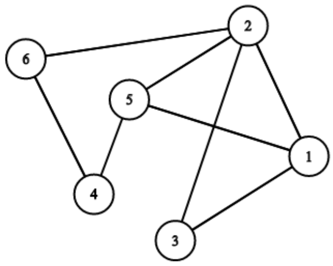
Componentele conexe ale lui G_1 sunt subgrafurile induse de mulțimile de noduri: {1, 3, 5, 7, 8, 9, 10, 11, 12}, {2} respectiv {4, 6}.

Termenul de maximal indică, de exemplu, că submulțimea de noduri formată din {1, 3, 5, 7, 9, 10, 11, 12} nu formează o componentă conexă pentru că prin adăugarea nodului 8 (și a muchiei corespunzătoare) se obține un subgraf care este în continuare conex, dar cu mai multe noduri.

Grafuri hamiltoniene și euleriene

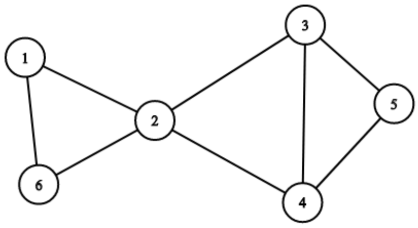
Graf hamiltonian

Un graf în care există un ciclu elementar care să conțină toate nodurile se numește graf hamiltonian. Un ciclu cu o astfel de proprietate se numește ciclu hamiltonian.



Graful din figura de mai sus este hamiltonian. Un ciclul elementar care conține toate nodurile este: 4, 5, 1, 3, 2, 6, 4.

Trebuie să reținem amănuntul că trebuie să avem ciclu elementar. Așadar, graful de mai jos nu este hamiltonian chiar dacă se poate găsi ciclu care să viziteze toate nodurile. Însă, un astfel de ciclu nu poate să atingă toate nodurile fără a-l folosi pe 2 de două ori (deci nu este elementar).



La categoriile speciale de grafuri se pune problema identificării unor rezultate echivalente (de exemplu ca în cazul grafurilor bipartite, unde avem garanția că un graf este bipartit **dacă și numai dacă** orice ciclu are lungime pară). Din păcate în cazul grafurilor hamiltoniene nu există un astfel de rezultat pentru cazul general. Astfel, algoritmii de detectare sunt de tip backtracking, generând toate permutările mulțimii de noduri și testând pentru fiecare în parte dacă ordinea reprezintă un ciclu.

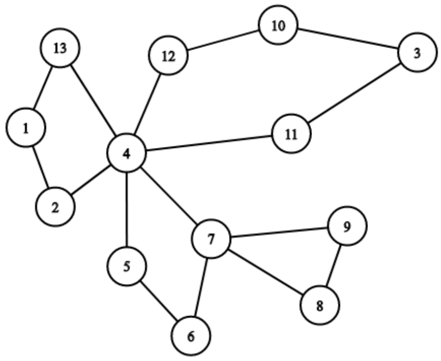
Sunt însă condiții suficiente care asigură că anumite categorii de grafuri sunt hamiltoniene. Iată un rezultat util:

Un graf în care fiecare nod are gradul cel puțin egal cu (n+1)/2 este cu siguranță hamiltonian. Pe un astfel de graf îl mai numim și dens, el având foarte multe muchii. Atenție că această condiție este doar suficientă, nefiind și necesară, spre exemplu graful “poligon” este hamiltonian dar toate nodurile au gradul 2, valoare care poate fi foarte mică în raport cu n.

Intervine și noțiunea de lanț hamiltonian: un lanț elementar în care apar toate nodurile. Graful din figura de mai sus, care nu este hamiltonian, conține totuși un lanț hamiltonian: 6, 1, 2, 3, 5, 4.

Graf eulerian

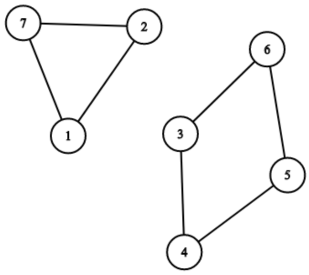
Un graf care conține un ciclu care să traverseze toate muchiile se cheamă graf eulerian. Ciclul nu trebuie să fie elementar. În fond, această noțiune studiază posibilitatea de a traversa o singură dată fiecare muchie și de a ne întoarce în locul de pornire, pe când noțiunea de graf hamiltonian studiază posibilitatea de a atinge exact o dată fiecare nod și de a ne întoarce în locul de pornire.



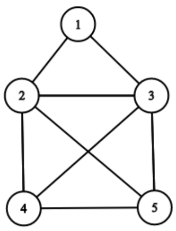
Graful din figura de mai sus este eulerian. Un ciclu eulerian este: `1, 2, 4, 5, 6, 7, 8, 9, 7, 4, 11, 3, 10, 12, 4, 13, 1`.

În acest caz avem și un rezultat care asigură o ***condiție necesară și suficientă ca un graf să fie eulerian: gradele tuturor nodurilor să fie pare și toate muchiile să fie în aceeași componentă conexă.***

Nu este necesar ca graful să fie conex, pot fi și noduri izolate, esențial este ca toate muchiile să fie în aceeași componentă conexă. Dacă nu impunem să avem muchiile în aceeași componentă conexă ci doar ca toate gradele să fie pare, ratăm situații ca aceea din figura următoare unde, evident, graful nu este eulerian.



Este definită și noțiunea de **lanț eulerian** (lanț simplu, care nu e ciclu, și care să conțină toate muchiile grafului). Evident că un graf care are doar lanț eulerian (fără a fi ciclu) nu este graf eulerian. Iată un exemplu:



Poate v-ați provocat colegul de bancă la următorul joc: poți desena figura de mai jos (considerând că nodurile - cercurile - sunt punctiforme) fără să ridici creionul de pe foaie, fără să tragi de două ori aceeași linie și să revii în punctul de plecare? De multe ori, repede se răspunde DA :D. La o analiză mai atentă a condițiilor impuse observăm că acestea cer de fapt să trasăm un ciclu eulerian. Ne dăm seama imediat că nu se poate pentru că avem noduri de grad impar (`4`, `5`). Există însă în acest graf un lanț eulerian: `4, 5, 2, 4, 3, 1, 2, 3, 5`. Adică pare că reușim să facem figura, fără să ridicăm pixul și fără să trasăm vreo linie a doua oară, însă nu îndeplinim ultima condiție: să revenim de unde am plecat.

Observăm că lanțul pornește dintr-un nod de grad impar și se termină în celălalt nod de grad impar. Așadar, grafurile care admit lanț eulerian care nu e și ciclu trebuie să aibă exact două noduri de grad impar.

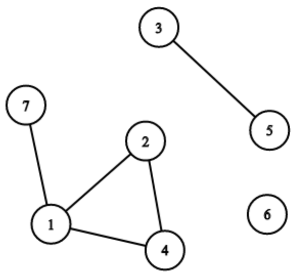
Reprezentarea (memorarea) grafurilor neorientate

Sunt diverse modalități de a stoca informații despre un graf. Aici vom studia două: matricea de adiacență (cu aplicabilitate mai degrabă din punct de vedere teoretic) și listele de vecini (modalitatea cel mai des folosită în practică).

Matricea de adiacență

Este o matrice pătratică, de dimensiune `nXn` în care elementele au semnificația:

$a_{i,j}$ =	`1`, dacă există muchie între la `i` la `j`
	`0`, dacă nu există muchie între la `i` la `j`

Graf	Matricea de adiacență
	0 1 0 1 0 0 1 1 0 0 1 0 0 1 0 0 0 0 1 0 0 1 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0

Iată câteva observații / proprietăți ale matricei de adiacență în cazul grafurilor neorientate.

- Este simetrică față de diagonala principală ($a_{i,j} = a_{j,i}$). Asta pentru că muchia `i,j` este aceeași cu muchia `j,i`.
- Pe diagonala principală toate elementele sunt `0` (nu putem avea muchie de la un nod la el însuși).
- Numărul de valori `1` de pe linia `i` (și totodată de pe coloana `i`, matricea fiind simetrică) reprezintă gradul nodului `i`.
- Indicii coloanelor unde pe linia `i` se află valoarea `1` reprezintă vecinii nodului `i`.
- Observăm că la matrice de adiacență diferite corespund grafuri diferite. În plus, matricea de adiacență nu dă informații despre poziția nodurilor în plan (spațiu) ceea ce în teoria grafurilor știm că nu este relevant.
- Întrucât matricea de adiacență este simetrică față de diagonala principală, sunt relevante doar elemetele din unul dintre triunghiurile delimitate de diagonala aceasta. Să îl considerăm, spre exemplu, pe cel de sub ea. Avem acolo un element pe linia a doua ($a_{2,1}$), două pe linia a treia ($a_{3,1}$ și $a_{3,2}$) ... $n-1$ elemente pe ultima linie. Sunt în total $1+2+ \; ... \; +(n-1) = n\cdot(n-1)/2$ elemente relevante care pot fi oricare dintre ele `0` și `1`. Știm că pe un șir cu `k` poziții binare avem 2^k posibilități de a pune `0` și `1`, în cazul nostru avem $2^{n\cdot(n-1)/2}$ variante de a construi o matrice de adiacență deci ajungem și astfel la formula, dedusă mai sus în material, pentru numărul total de grafuri cu `n` noduri.

Listele de vecini (de adiacență)

Se păstrează, pentru fiecare nod o structură liniară în care se memorează doar vecinii săi. Pentru graful reprezentat mai sus cu matrice de adiacență, listele de vecini sunt:

L_1	2, 4 7
L_2	4, 1
L_3	5
L_4	1, 2
L_5	3
L_6	
L_7	1

În acest caz, gradul nodului `i` este dat de numărul de elemente ale listei i iar vecinii nodului `i` sunt chiar elementele listei `i`.

Iată argumentele pentru care în practică listele de vecini sunt preferate.

- **Memoria**: Numărul de locații necesare pentru matricea de adiacență este n^2, iar în cazul listelor de vecini se ajunge la `n·(n-1)` doar în cazul grafului complet. Imaginați-vă că sunt mereu întâlnite în practică grafuri cu, să spunem, `100000` de noduri și `300000` – `400000` de muchii (exemplul care îmi vine întâi în minte este cu o rețea de drumuri între localități și știm că fiecare localitate este legată doar cu un număr mic de localități vecine în mod direct, așadar putem avea număr mare de localități – noduri – dar numărul de muchii să nu meargă către maxim). Într-un astfel de caz este imposibil de folosit matricea de adiacență, pe când listele de vecini ar asigura număr de locații necesare de ordin `m` (`2·m` mai exact în cazul grafurilor neorientate).
- **Timpul**: În lucrul cu grafuri, operația de bază este identificarea vecinilor unui nod. La o reprezentare prin matrice de adiacență acest lucru implică `n` pași pentru un nod (parcurgerea liniei sale din matricea de adiacență) și dacă s-ar face operația pentru toate nodurile (lucru necesar la parcurgerea grafurilor DFS, BFS) s-ar ajunge la timp de ordin n^2. În cazul memorării cu liste de vecini, la fiecare nod se vizitează exact vecinii săi (găsiți în listă), deci în total, pentru toate nodurile se ajunge la `2·m` pași.

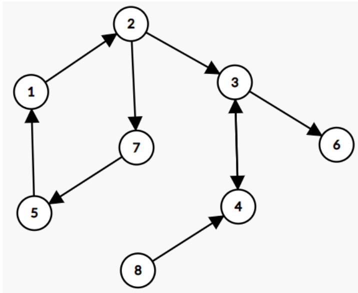
Întrucât în cazul numărului mic de muchii atât ca memorie cât și ca timp valoarea dominantă poate rămâne numărul de noduri, spune că în cazul listelor de vecini atât timpul de executare cât și memoria necesară sunt de ordinul `n+m`.

Grafuri orientate

În cazul în care toate muchiile au asociate săgeți, graful este orientat. Deci, subliniem, nu se pune problema ca unele muchii să aibă asociate săgeți și altele nu. Dacă unei muchii dorim să nu îi asociem săgeți (să o considerăm în ambele sensuri) punem de fapt două muchii, una orientată într-un sens și una în celălalt.

De regulă, în cazul grafurilor orientate este permisă prezența mai multor muchii între aceleași două noduri și în același sens. Se spune că avem de-a face cu un multigraf. Este permisă, de asemenea și muchie între un nod și el însuși, aceasta chemându-se buclă.

În continuare, în acest material vom considera că grafurile orientate cu care lucrăm nu sunt multigrafuri și nu au bucle.



Mai sus avem desenat un graf orientat care nu este multigraf și care nu are bucle (nu este multigraf pentru că între 3 și 4 cele două muchii sunt distincte, având orientări diferite).

Iată câteva diferențe, ca terminologie, între grafurile neorientate și cele orientate.

(*) Se spune **vârf** în loc de nod;

(*) Se spune **arc** în loc de muchie;

(*) Noțiunea de grad se separă. Avem **grad exterior** al unui vârf ca fiind numărul de arce care ies din acel vârf și **grad interior** ca fiind numărul de arce care intră în vârf. Vom nota d^+ gradul exterior și d^- gradul interior.

Avem astfel:

- $d^+(4) = 1$
- $d^-(4) = 2$

Subliniem aici un rezultat important: **într-un graf orientat suma gradelor exterioare este egală cu suma gradelor interioare și egală totodată cu numărul de arce**.

$$d^+(1) + d^+(2) + ... + d^+(n) = d^-(1) + d^-(2) + ... + d^-(n) = m$$

Justificăm simplu acest rezultat dacă ne gândim că fiecare arc contribuie cu 1 la suma gradelor exterioare și cu 1 la suma gradelor interioare.

(*) Avem aici noțiunea de **drum** (în locul celei de lanț de la grafurile neorientate). Drum reprezintă o secvență de vârfuri cu proprietatea că oricare două vecine formează un arc (în sensul de parcurgere).

- 1, 2, 3, 6 este un drum (chiar **elementar**, gândindu-ne că nu se repetă vârfuri);
- 1, 2, 3, 4, 3, 6 este un drum neelementar;
- 8, 4, 3, 2, 7 nu este drum pentru că nu avem arc de la 3 la 2 (avem de la 2 la 3);

(*) Un drum simplu, închis se cheamă **circuit** (echivalent pentru ciclu de la grafuri neorientate).

(*) Ca și la grafurile neorientate definim **lungimea** unui drum (circuit) ca fiind **numărul de arce**.

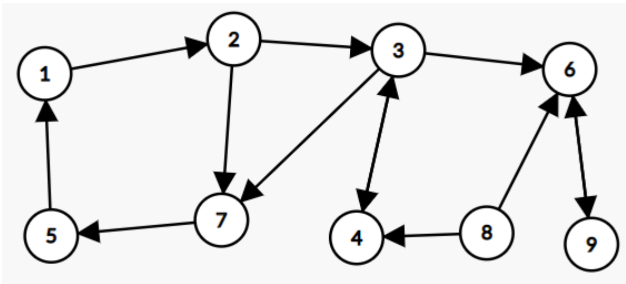
(*) Se definesc în mod similar noțiunile de **graf parțial** și de **subgraf**. Graf parțial se obține păstrând toate vârfurile și o submulțime a mulțimii arcelor iar subgraf se obține păstrând o submulțime a mulțimii vârfurilor și doar arcele incidente lor.

(*) Un alt rezultat important este: cu n vârfuri se pot forma $4^{C_n^2}$. Ca și în cazul grafurilor neorientate sunt multiple feluri în care putem justifica asta. Putem ține minte ușor formula gândindu-ne, de exemplu, astfel:

avem C_n^2 moduri de a selecta o pereche de arce distincte. Între o astfel de pereche (a, b) avem 4 moduri de a alege cum trasăm arce: doar de la a la b, doar de la b la a, în ambele sensuri și deloc.

(*) avem aici de a face cu noțiunea de **tare conexitate** (și , legată de ea, noțiunea de **componentă tare conexă**).

Un graf orientat este tare conex dacă între oricare două vârfuri există drum, în ambele sensuri, Noțiunea de componentă tare conexă se definește ca în cazul noțiunii de componentă conexă la grafurile orientate: un subgraf tare conex și maximal cu această proprietate (maximal în sensul că dacă s-ar extinde cu vârfuri subgraful, acesta n-ar mai fi tare conex).



Graful de mai sus nu este tare conex. Unul dintre motive este că nu sunt drumuri care ajung în 8 (chiar dacă observăm că din 8 se poate ajunge în oricare dintre celelalte vârfuri).

Totodată avem trei componente tare conexe:

- Subgraful format cu 1, 2, 3, 4, 5, 7;
- Subgraful format din 6, 9;
- Subgraful format din 8;

(*) Modalități de reprezentare

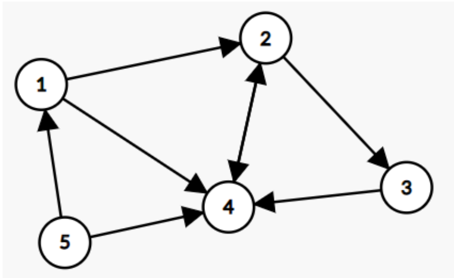
Pentru grafurile orientate putem utiliza cele două modalități de reprezentare prezentate în cazul grafurilor neorientate: matricea de adiacență și listele de vecini.

Matricea de adiacență

- Este pătratică, de dimensiune nXn și în care avem:

<table>
<tr><td rowspan="2">$a_{i,j}$ =</td><td>1, dacă există arc de la i la j</td></tr>
<tr><td>0, în caz contrar</td></tr>
</table>

- Conform definiției de mai sus, nu mai este simetrică față de diagonala principală;
- Numărul de valori 1 de pe linia i reprezintă gradul exterior al lui i;
- Numărul de valori 1 de pe coloana i reprezintă gradul interior al lui i;



Pentru graful de mai sus avem următoarea matrice de adiacență:

```
0 1 0 1 0
0 0 1 1 0
0 0 0 1 0
0 1 0 0 0
1 0 0 1 0
```

Liste de vecini

În cazul de față listele de vecini sunt de regulă liste de succesori. În funcție de problemă programatorul poate alege să folosească separat și liste de predecesori.

Listele de succesori pentru graful a cărui matrice de adiacență este scrisă mai sus, sunt:

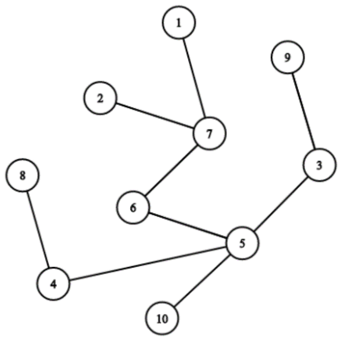
L_1	2, 4
L_2	4, 3
L_3	4
L_4	2
L_5	1, 4

Trebuie subliniat că deseori se face (și se acceptă) abuz de limbaj când vorbim de noțiuni legate de grafuri: spunem noduri, muchii, cicli și în cazul grafurilor orientate.

Arbori

La pornirea discuției despre arbori deseori se întâmplă ca elevii să se gândească la ceva ierarhizat pe nivele. Și este adevărat, dacă ne gândim la arborii cu rădăcină. Însă vom porni discuția de la arbori oarecare, care nu au nicio legătură cu așezarea.

Astfel, conform definiției, un **arbore este un graf conex și fără cicli**.



Avem mai sus un arbore oarecare.

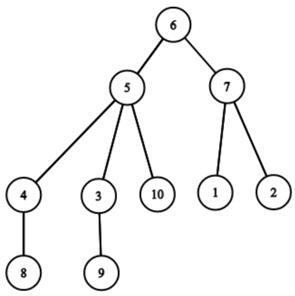
Alături ce cele două proprietăți din definiție (graf conex și fără cicli), alte `4` prezentate în continuare reprezintă rezultate foarte utile și des folosite:

- **Un arbore cu `n` noduri are `n-1` muchii**. Putem justifica simplu acest lucru prin inducție, dacă plecăm de la un arbore cu două noduri (și o muchie), orice nod adăugat vine împreună cu o muchie (cea cu care îl legăm de arborele format deja anterior);
- **Un arbore este conex la limită**, adică oricare muchie am elimina dintre cele existente, se pierde conexitatea;
- **Un arbore este fără cicli la limită**, adică orice muchie nouă am trage între nodurile existente, se formează un ciclu;
- Într-un arbore **există lanț elementar unic între oricare două noduri**.

Arbori cu rădăcină

Modul cel mai simplu de a explica ce înseamnă arbore cu rădăcină este să ne gândim că pornim de la un arbore oarecare și îl "agățăm" într-un nod, imaginându-ne că muchiile și nodurile celelalte "cad".

De exemplu, "agățând" arborele anterior în nodul `6`, acesta ar arăta:



Acest arbore nu îl vom mai privi acum ca pe unul oarecare: el este **arbore cu rădăcină**.

Terminologie:

- Nodul din vârf se cheamă **rădăcină** (`6` în exemplul nostru);
- Descendenții direcți ai unui nod se cheamă **fii** ai acelui nod (`5` și `7` sunt fii ai lui `6` iar `4`, `3`, `10` sunt fii ai lui `5`);
- Dacă `i` este fiu al lui `j` atunci spunem că `j` este **tată** al lui `i`;

- Toate nodurile unui arbore au câte un tată, mai puțin rădăcina, care nu are;
- Nodurile care nu au fii se cheamă **frunze** (8, 9, 10, 1 și 2 din exemplul nostru);
- Uneori la fii li se mai spune **descendenți** direcți ai unui anume nod iar termenul descendenți (fără direcți) se referă la toate nodurile din subarborele unui anume nod (de exemplu 4, 3, 10, 8, 9 sunt descendenți ai lui 5);
- **Ascendenți** ai unui nod sunt considerate nodurile aflate pe lanțul elementar unic de la acel nod la rădăcină (de exemplu, ascendenții lui 3 sunt 5 și 6);
- **Înălțimea** unui arbore este reprezentată de lungimea maximă a unui lanț de la rădăcină la o frunză; în arborele de mai sus avem înălțimea 3, obținută considerând lanțul spre frunza 8 sau pe cel spre frunza 9;
- Nodurile aflate la aceeași distanță față de rădăcină se află pe același **nivel**. De exemplu, rădăcina (nodul 6) este pe nivelul 0, iar nodurile 4, 3, 10, 1 și 2 sunt pe nivelul 2.

Reprezentarea arborilor cu rădăcină.

În general, metodele folosite la reprezentarea grafurilor se pot utiliza și la cazurile particulare de grafuri care sunt arborii. De exemplu reprezentarea cu liste de vecini a unui arbore oarecare este o alegere foarte bună, făcându-se economie de memorie în comparație cu reprezentarea arborelui prin matrice de adiacență (arborele are număr de muchii de ordin n, mai exact n-1 iar matricea de adiacență va avea cele mai multe dintre cele n^2 elemente, egale cu 0).

Pentru arborii cu rădăcină există și o modalitate de reprezentare care le pune foarte bine în evidență structura. Aceasta se folosește de faprul că fiecare nod are un singur tată, păstrând doar un vector în care pe poziția i se află tatăl nodului i (sau un semn, de regulă 0, decă i este rădăcina). O astfel de structură se cheamă **vector de tați**.

Pentru arborele cu rădăcină de mai sus, vectorul de tați este:

Indice (nod)	1	2	3	4	5	6	7	8	9	10
T[nod]	7	7	5	5	6	0	6	4	3	5

Iată cum extragm din vectorul de tați informații despre arbore.

- O pereche de forma (i, T[i]), cu T[i] nenul, reprezintă o muchie. Obținem astfel cele n-1 muchii ale arborelui.
- Este o singură valoare 0 în vector, cea corespunzătoare poziției care indică rădăcina.
- Elementele care nu apar în vectorul de tați sunt frunzele.
- Un cod de forma: `while (x) {scrie x; x = T[x];}` oferă lanțul de la nodul x la rădăcină.

Probleme rezolvate

Parcurgerea lor are un rol deosebit în aprofundarea și legarea noțiunilor despre grafuri.

1. Arătați că următorul șir de numere nu poate reprezenta valorile gradelor nodurilor unui graf neorientat: 2, 3, 1, 4, 3, 2.

Rezolvare

Suma acestor valori este un număr impar iar noi știm că suma gradelor nodurilor unui graf este o valoare pară (dublul numărului de muchii).

2. Se consideră un graf neorientat cu 20 de noduri și 33 de muchii. Care este numărul maxim de componente conexe pe care poate să le aibă un astfel de graf?

Rezolvare

Ne imaginăm că pornim cu 20 de noduri izolate, adică 20 de componente conexe. Între aceste noduri vom duce cele 33 de muchii una câte una. Pot să apară două cazuri:

- La muchia curentă, dacă ea este dusă între două noduri din aceeași componentă conexă, numărul de componente conexe rămâne nemodificat.
- Dacă o ducem între noduri aflate în acel moment în componente conexe diferite, vom reuni practic cele două componente conexe în una singură, deci numărul de componente conexe scade cu 1.

Așadar, dorim sa ducem un număr maxim de muchii între noduri aflate la momentul respectiv în aceeași componentă conexă.

Dacă la un moment dat avem un subgraf complet cu x noduri (și restul noduri izolate – pornirea este pentru $x=1$), pentru a duce următoarea muchie avem două variante:

- Ducem muchia între două noduri izolate;
- Ducem muchia între un nod izolat și un nod din subgraful complet;

În ambele moduri numărul de componente conexe scade cu 1. Dacă alegem însă să ducem ca în primul caz, vom mai forma un subgraf complet (cu două noduri) deci următoarea alegere nu se poate face în cadrul aceleiași componente conexe (implicit la ea va scădea iarăși numărul de componente conexe) pe când dacă alegem să procedăm ca în cazul 2, următorele $x-1$ muchii se duc între nodul ales și celelalte noduri din componenta conexă, deci fiecare dintre ele este dusă fără să scădem numărul de componente conexe.

În concluzie, încercăm să facem un subgraf complet cât mai mare cu muchiile avute la dispoziție.

Cea mai mare valoare $x\cdot(x-1)/2$ mai mică sau egală cu 33 este 28 (pentru $x=8$). Așadar, consumăm 28 de muchii cu 8 noduri între care ducem muchii în toate modurile posibile. Alegem apoi al 9-lea nod pe care îl unim cu 5 din cele 8, consumând toate muchiile și obținând o componentă conexă cu 9 noduri și încă 11 noduri izolate. În total, deci, 12 componente conexe.

3. Considerăm un graf cu 20 de noduri și 33 de muchii. Care este numărul maxim de noduri izolate pe care le poate avea avesta ?

Rezolvare

Facem un raționament identic cu cel de la problema anterioară. Rărpuns: 11.

4. Se consideră un graf conex cu 100 de noduri și 1000 de muchii. Care este numărul de muchii care trebuie eliminate așa încât graful obținut să fie arbore.

Rezolvare

Un arbore cu 100 de noduri are 99 de muchii. Răspunsul este 1000-99 = 901.

5. Care este gradul maxim posibil pentru un nod al unui arbore cu `20` de noduri.

Rezolvare

Arborele are un nod unit cu toate celelalte. Așadar 19 este răspunsul. Acesta se mai numește graf stea.

6. Care este gradul minim posibil pentru un nod al unui arbore cu `20` de noduri.

Rezolvare

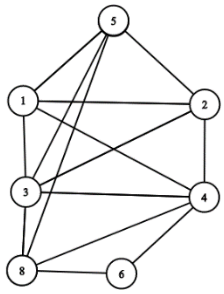
Răspunsul este `1`. În orice arbore are cel puțin un nod de grad `1`, altfel sigur ar apărea cicli.

7. Care este numărul de muchii ce trebuie eliminate dintr-un graf complet cu `10` noduri așa încât acesta să fie arbore ?

Rezolvare

Graful complet cu `10` noduri are `10·9/2 = 45` de muchii. Un arbore cu `10` noduri are `9` muchii. Răspunsul este `45-9 = 36`.

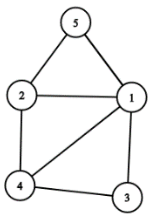
8. Pentru graful din figură, care este numărul minim de muchii ce trebuie duse încât el să devină eulerian ?



Rezolvare

Graful este conex și singurele noduri de grad impar sunt `3` și `4`. Ideal era să nu fie muchie între ele și atunci am fi dus această muchie și deveneau toate nodurile de grad par, deci rezolvam problema cu rezultat `1`. Însă noi avem muchie, deci `1` nu poate fi rezultatul. Acum ne gândim să găsim un nod intermediar `x`, să ducem muchie între `3` și `x` apoi între `x` și `4`. Însă toate nodurile sunt legate fie de `3` fie de `4`. Ne dăm seama că putem generalizam adică dacă am duce un lanț elementar de la `3` la `4`, cu diverse noduri intermediare, la nodurile din interiorul lanțului gradul ar crește cu `2`, deci nu li s-ar modifica paritatea gradului iar la cele două extremități, care ne interesează pe noi, gradul devine par. Deci acum o să căutăm lanț de la `3` la `4` cu doi intermediari, deci de lungime `3` și găsim: `3`, `6`, `5`, `4`. Așadar răspunsul este `3`.

9. Care este numărul de cicluri elementare ale grafului din figura de mai jos ?



Rezolvare

În astfel de situații se recomandă să numărăm ciclii după lungimi: de lungime 3, avem trei cicli (1,2,5,1), (1,2,4,1) și (1,3,4,1), de lungime 4 avem doi cicli (1,5,2,4) și (1,2,4,3) și unul de lungime 5 (1,5,2,4,3,1). În total 6.

10. Se consideră un graf neorientat cu 5 noduri și 9 muchii. Care dintre următoarele poate fi șirul gradelor nodurilor grafului?

4, 2, 6, 4, 2	Nu, pentru că fiind 5 noduri gradul maxim al vreunuia poate fi până la 4, ori în acest șir avem și valoarea 6.
2, 2, 1, 2, 2	Nu, pentru că suma gradelor ar fi un număr impar, lucru imposibil.
4, 4, 4, 4, 4	Nu, pentru că suma gradelor este 20, și ea ar trebui să fie dublul numărului de muchii, deci 18.
4, 3, 3, 4, 4	Da.

11. Câte lanțuri elementare distincte, cu 4 noduri, are un graf complet cu 10 noduri?

Rezolvare

Avem combinări de 10 luate câte 4 moduri de a alege componența lanțului. La fiecare alegere avem 4 factorial moduri de a așeza în ordine cele 4 elemente alese. Dar la fiecare astfel de permutare există și cea oglindită care reprezintă același lanț. Avem deci: $C_{10}^{4} * \frac{4!}{2}$ soluții.

12. Care dintre următorii reprezintă vectori de tați valizi pentru un arbore cu rădăcină?

0 1 1 1 3 3 3 3	Da, este arborele cu rădăcina 1, acesta are fii pe 2, 3, 4 iar 3 are fii celelalte noduri.
0 1 1 0 2 2	Nu, întrucât nu putem avea mai mult de o rădăcină, deci vectorul de tați nu poate avea două valori 0.
0 3 2 1 4 4	Nu. Aici avem o singură valoare 0, dar apare următoarea situație: 3 are ca tata pe 2 și 2 are ca tata pe 3. Imposibil.

13. Considerăm un arbore cu rădăcină în care fiecare nod poate avea 0, 1 sau 2 fii. Știind că acest arbore are exact 100 de noduri cu 2 fii, câte frunze are el ?

Rezolvare

De multe ori se întâmplă ca primul răspuns primit să fie: “nu sunt informații suficiente”. Raționamentul care duce la soluție este următorul: Dacă avem x frunze și pe una dintre ele o anulăm, ducându-i un singur fiu, apare acel fiu ca frunză și dispare cea aleasă deci numărul de frunze rămâne constant. Dacă pe una o anulăm și îi ducem doi fii, numărul de frunze crește cu 1 (apar două, dispare una). Deci dacă ne imaginăm că pornim de la un singur nod (pe care în spiritul problemei îi putem considera frunză), avem o frunză și 0 noduri cu doi fii. Cum la fiecare nod cu doi fii apare o frunză în plus, rezultatul final va fi 101. Observăm că acesta nu depinde de numărul total de noduri ale grafului.