Un vocabular al teoriei grafurilor

Definiția unui graf

Un ***graf*** este o pereche $\boldsymbol{G = (V(G),E(G))}$, unde $\boldsymbol{V(G)}$ este o mulțime finită nevidă, iar $\boldsymbol{E(G)}$ este o submulțime a mulțimii $\boldsymbol{P_2(V(G))}$ a părților cu două elemente ale lui $\boldsymbol{V(G)}$.

$\boldsymbol{V(G)}$ se numeşte ***mulțimea vârfurilor*** grafului $\boldsymbol{G}$ şi numărul său de elemente, $\boldsymbol{|V(G)|}$, este ***ordinul grafului G***; $\boldsymbol{E(G)}$ este ***mulțimea muchiilor*** grafului $\boldsymbol{G}$ şi numărul său de elemente, $\boldsymbol{|E(G)|}$, este ***dimensiunea*** grafului $\boldsymbol{G}$. Atunci când nu există posibilitatea confuziilor, vom nota simplu, $\boldsymbol{G = (V, E)}$.

Dacă $\boldsymbol{e = \{u, v\} \in E(G)}$ este o muchie a grafului $\boldsymbol{G}$ vom nota $\boldsymbol{e = uv}$ (pentru simplificarea scrierii) şi vom spune că: muchia ***e unește*** vârfurile $\boldsymbol{u}$ şi $\boldsymbol{v}$; vârfurile $\boldsymbol{u}$ şi $\boldsymbol{v}$ sunt ***adiacente*** în $\boldsymbol{G}$; muchia $\boldsymbol{e}$ este ***incidentă*** cu vârfurile $\boldsymbol{u}$ şi $\boldsymbol{v}$; vârfurile $\boldsymbol{u}$ şi $\boldsymbol{v}$ sunt ***vecine*** în $\boldsymbol{G}$; vârfurile $\boldsymbol{u}$ şi $\boldsymbol{v}$ sunt ***extremitățile*** muchiei $\boldsymbol{e}$.

Dacă $\boldsymbol{v \in V(G)}$, atunci mulțimea $\boldsymbol{N_G(v) = \{w \mid w \in V(G), vw \in E(G)\}}$ se numeşte ***vecinătatea*** vârfului $\boldsymbol{v}$ în $\boldsymbol{G}$. Vom mai nota $\boldsymbol{N_G(v) = \Gamma_G(v)}$. Remarcăm faptul că graful $\boldsymbol{G}$ poate fi definit (în mod echivalent) ca o pereche $(\boldsymbol{V(G), \Gamma_G})$ unde

$$\boldsymbol{\Gamma_G : V(G) \to P(V(G))}$$

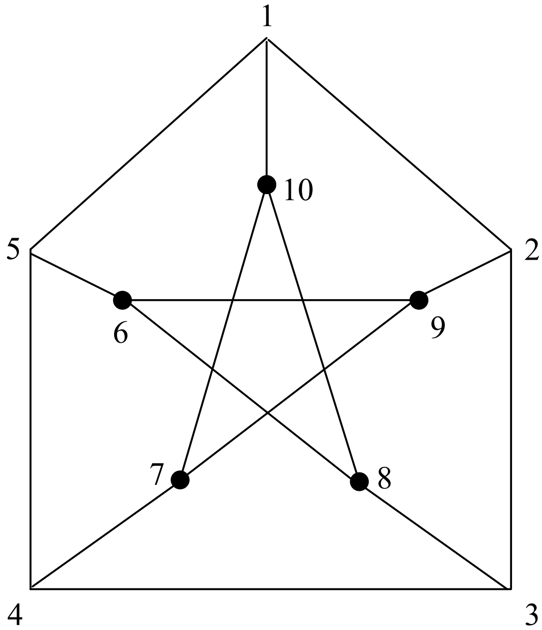
asociază fiecărui vârf vecinătatea sa (astfel încât $\boldsymbol{v \notin \Gamma_G(v)}$ şi dacă $\boldsymbol{w \in \Gamma_G(v)}$, atunci $\boldsymbol{v \in \Gamma_G(w)}$, pentru orice vârfuri $\boldsymbol{v}$ şi $\boldsymbol{w}$).
Două muchii $\boldsymbol{e}$ şi $\boldsymbol{e'}$ care au o extremitate comună se numesc ***adiacente***.

O ***mulțime independentă de vârfuri*** sau ***mulțime stabilă*** în $\boldsymbol{G}$ este o mulțime $\mathbf{S} \subseteq \boldsymbol{V(G)}$ de vârfuri cu proprietatea că $\boldsymbol{P_2(S) \cap E(G) = \varnothing}$ (adică o mulțime de vârfuri neadiacente două câte două). Cardinalul maxim al unei mulțimi stabile se numeşte ***numărul de stabilitate*** sau ***numărul de independență*** al grafului $\boldsymbol{G}$ şi se notează cu $\alpha(\boldsymbol{G})$.

O ***mulțime independentă de muchii*** sau ***cuplaj*** în graful $\boldsymbol{G}$ este o mulțime de muchii neadiacente două câte două. Cardinalul maxim al unei mulțimi independente de muchii în $\boldsymbol{G}$ se numeşte ***numărul de muchie-independență*** al grafului $\boldsymbol{G}$ şi se notează $\nu(\boldsymbol{G})$.

Intuitiv, un graf $\boldsymbol{G = (V(G), E(G))}$ poate fi reprezentat (după cum sugerează şi numele său) cu ajutorul unei figuri plane formată dintr-o mulțime de cerculețe îngroşate aflată în corespondență cu mulțimea de vârfuri $\boldsymbol{V(G)}$, două cerculețe fiind unite printr-o curbă simplă dacă şi numai dacă, perechea de vârfuri corespunzătoare lor este o muchie a grafului $\boldsymbol{G}$.

De exemplu, graful $\boldsymbol{G = (V, E)}$ unde $\boldsymbol{V = \{1, 2, \ldots, 10\}}$ şi $\boldsymbol{E = \{12, 23, 34, 45, 56, 51, 69, 68, 710, 79, 810, 92, 10\,1, 74, 83\}}$ poate fi reprezentat de



Să observăm că în acest graf:
Mulţimea de vârfuri $\boldsymbol{\{10, 6, 2, 4\}}$ este stabilă şi este de cardinal maxim, deci $\boldsymbol{\alpha(G) = 4}$; mulţimea de muchii $\boldsymbol{\{110, 29, 38, 47, 56\}}$ formează un cuplaj de cardinal maxim, deci $\boldsymbol{\nu(G) = 5}$.

Dacă $\boldsymbol{G = (V(G), E(G))}$ este un graf şi $\boldsymbol{p \in N^*}$, se numeşte ***p-colorare*** a (*vârfurilor*) lui $\boldsymbol{G}$ o aplicaţie $\boldsymbol{c : V(G) \to \{1, \ldots, p\}}$ cu proprietatea că $\boldsymbol{c^{-1}(i)}$ este o mulţime stabilă în $\boldsymbol{G}$, $(\forall)$ $\boldsymbol{i \in \{1, \ldots, p\}}$ (remarcăm că, din definiţia mulţimilor stabile, $\varnothing$ este o mulţime stabilă). ***Numărul cromatic*** al grafului $\boldsymbol{G}$, notat $\chi(\boldsymbol{G})$, este cea mai mică valoare a lui $\boldsymbol{p \in N^*}$ pentru care $\boldsymbol{G}$ admite o $\boldsymbol{p}$-colorare.

O ***p-colorare a muchiilor*** lui $\boldsymbol{G}$ este o aplicaţie $\boldsymbol{c : E(G) \to \{1, \ldots, p\}}$ cu proprietatea că $\boldsymbol{c^{-1}(i)}$ este un cuplaj al lui $\boldsymbol{G}$, $(\forall)$ $\boldsymbol{i \in \{1, \ldots, p\}}$. ***Indicele cromatic*** al grafului $\boldsymbol{G}$, notat $\chi'(\boldsymbol{G})$, este cea mai mică valoare a lui $\boldsymbol{p \in N^*}$ pentru care $\boldsymbol{G}$ admite o $\boldsymbol{p}$-colorare a muchiilor.

Două grafuri, $\boldsymbol{G = (V(G), E(G))}$ şi $\boldsymbol{H = (V(H), E(H))}$ se numesc ***izomorfe***, şi notăm aceasta prin $\boldsymbol{G \cong H}$, dacă există o bijecţie $\boldsymbol{\varphi : V(G) \to V(H)}$ cu proprietatea că aplicaţia $\boldsymbol{\psi : E(G) \to E(H)}$, definită pentru orice $\boldsymbol{uv \in E(G)}$ prin $\boldsymbol{\psi(uv) = \varphi(u)\varphi(v)}$, este o bijecţie (deci, două grafuri sunt izomorfe dacă există o bijecție între mulțimile lor de vârfuri care induce o bijecție între mulțimile lor de muchii).

Variații în definiția unui graf

a) Dacă în definiția unui graf, se consideră $\boldsymbol{E(G)}$ o multimulțime pe $\boldsymbol{P_2(V(G))}$, adică este dată o funcție

$$m : P_2(V(G)) \to N,$$

se obține noțiunea de ***multigraf***. Un element $\boldsymbol{e \in P_2(V(G))}$ cu $\boldsymbol{m(e) > 0}$ este muchie a multigrafului, ***simplă*** dacă $\boldsymbol{m(e) = 1}$, ***multiplă*** dacă $\boldsymbol{m(e) > 1}$. Oricărui multigraf $\boldsymbol{M}$ i se poate asocia un graf $\boldsymbol{G(M)}$, numit ***graful suport*** al lui $\boldsymbol{M}$, obținut prin înlocuirea fiecărei muchii multiple cu o singură muchie cu aceleaşi extremități.

b) Dacă în definiția unui graf se consideră $\boldsymbol{E(G)}$ ca o multimulțime pe mulțimea părților nevide cu cel mult două elemente ale lui $\boldsymbol{V(G)}$, atunci $\boldsymbol{G}$ se numeşte ***graf general*** sau ***pseudograf***. O muchie $\boldsymbol{e \in E(G)}$, $\boldsymbol{e = \{v\}}$ se numeşte ***buclă*** în vârful $\boldsymbol{v}$. Pictural pseudografurile se reprezintă la fel ca şi grafurile.

Pentru evitarea confuziilor, uneori grafurile – aşa cum le-am definit – se mai numesc şi ***grafuri simple***.

c) Un ***digraf*** este o pereche $\boldsymbol{D = (V(D), A(D))}$ unde $\boldsymbol{V(D)}$ este o mulțime finită nevidă (mulțimea vârfurilor digrafului $\boldsymbol{D}$), iar $\boldsymbol{A(D) \subseteq V(D) \times V(D)}$ este mulțimea ***arcelor*** digrafului $\boldsymbol{D}$. Dacă $\boldsymbol{a = (u, v)}$ este arc în $\boldsymbol{D}$, notăm $\boldsymbol{a = uv}$ şi spunem că $\boldsymbol{u}$ şi $\boldsymbol{v}$ sunt ***adiacente***; $\boldsymbol{a}$ este ***incident din u***; $\boldsymbol{a}$ este ***incident spre v***; ***u domină*** pe $\boldsymbol{v}$; $\boldsymbol{a}$ este ***incident*** cu ***u spre exterior*** ; $\boldsymbol{a}$ este ***incident*** cu ***v spre interior*** ; $\boldsymbol{u}$ este ***extremitatea inițială*** a lui $\boldsymbol{a}$ şi $\boldsymbol{v}$ este ***extremitatea finală*** a lui $\boldsymbol{a}$.

Pictural, digrafurile se reprezintă la fel ca şi grafurile, adăugând curbei care uneşte două cerculețe un sens pentru a preciza perechea de vârfuri corespunzătoare arcului desenat.

O pereche de arce de forma $\boldsymbol{vw}$ şi $\boldsymbol{wv}$ se numeşte ***pereche simetrică*** de arce. Dacă $\boldsymbol{D}$ este un digraf, ***inversul*** său $\boldsymbol{D'}$ este digraful obținut din $\boldsymbol{D}$ prin înlocuirea fiecărui arc $\boldsymbol{vw}$ cu opusul său $\boldsymbol{wv}$.

Dacă $\boldsymbol{D}$ este un digraf, atunci înlocuind fiecare arc cu mulțimea de vârfuri care îl formează, obținem, în general, un multigraf $\boldsymbol{M(D)}$. Graful suport al acestui multigraf se numeşte ***graful suport*** al grafului $\boldsymbol{D}$. Dacă $\boldsymbol{M(D)}$ este graf atunci $\boldsymbol{D}$ se numeşte ***graf orientat*** (poate fi gândit ca obținut prin "orientarea" fiecărei muchii a grafului $\boldsymbol{M(D)}$).

Un ***digraf complet simetric*** este un digraf în care fiecare pereche de vârfuri este unită prin exact o pereche de arce simetrice. Un ***turneu*** este un digraf în care orice două vârfuri sunt unite prin exact un arc.

d) Grafurile ***infinite*** se obțin prin înlăturarea condiției de finitudine a mulțimii de vârfuri şi (sau) muchii. Un graf ***G local finit*** este un graf finit în care $\boldsymbol{N_G(v)}$ este finită pentru orice vârf $\boldsymbol{v}$.

Grade

Dacă $\boldsymbol{G = (V, E)}$ este un graf şi $\boldsymbol{v \in V}$ un vârf al său, atunci ***valența*** sau ***gradul*** lui $\boldsymbol{v}$ în $\boldsymbol{G}$, notat $\boldsymbol{d_G(v)}$ sau $\boldsymbol{p_G(v)}$, este

$$|\{e \mid e \in E, e \text{ incidentă cu } v\}|.$$

Un vârf de grad **0** se numeşte ***izolat***; un vârf de grad **1** se numeşte ***pendant***. Dacă toate vârfurile lui ***G*** au aceeaşi valenţă ***p*** atunci ***G*** se numeşte ***graf p – valent*** sau ***p – regulat***. Un graf **0** – valent se numeşte ***graf nul***. Un graf **3** – valent se numeşte ***graf trivalent*** sau ***cubic***. Un exemplu de graf trivalent este graful lui ***Petersen***:

Concepte analoage se pot defini şi pentru digrafuri. Dacă ***v*** este un vârf al digrafului ***D*** atunci ***valenţa interioară*** sau ***gradul interior*** notat $\mathbf{p_{in}(v)}$ sau p_D^- sau d_D^-, este numărul arcelor incidente cu ***v*** spre interior; ***valenţa extrerioară*** sau ***gradul exterior*** notat $\mathbf{p_{out}(v)}$ sau
p_D^+ sau $\boldsymbol{d_D^+}$, este numărul arcelor incidente cu ***v*** spre exterior.

Subgrafuri

Un ***subgraf*** al grafului $G = (V(G), E(G))$ este un graf $H = (V(H), E(H))$ care satisface: $V(\mathrm{H}) \subseteq V(G)$ şi $E(\mathrm{H}) \subseteq E(G)$.

Dacă $V(\mathrm{H}) = V(G)$ atunci ***H*** se numeşte ***graf parţial*** al lui ***G***.

$A \subseteq V(G)$ atunci $[A]_G = (A, \mathrm{P}_2(A) \cap E(G))$ se numeşte ***subgraf indus*** în ***G*** de mulţimea de vârfuri ***A***. subgraful $[V(G) \setminus A]_G$ se notează $G - A$ şi este ***subgraful*** lui ***G obţinut prin îndepărtarea vârfurilor din A***; în particular, dacă $A = \{v\}$, atunci $G - \{v\}$ se notează $G - v$.

Dacă $E' \subseteq E(G)$ atunci $\langle E' \rangle_G = (V(G), E')$ este ***graful parţial secţionat de E' în G***. $G - E'$ este prin definiţie $\langle E(G) \setminus E' \rangle_G$, iar $G - e = G - \{e\}$ $(e \in E(G))$.

Concepte similare se pot defini în mod analog pentru multigrafuri, grafuri generale sau digrafuri.

Operaţii cu grafuri

Dacă $G = (V(G), E(G))$ este un graf, atunci:

- ***complementarul*** său este graful $\overline{G}$ cu $V(\overline{G}) = V(G)$ şi
$$E(\overline{G}) = P_2(V(G)) \setminus E(G).$$
- ***graful reprezentativ al muchiilor lui G*** este graful $L(G)$ cu $V(L(G)) = E(G)$ şi $E(L(G)) = \{ee' \mid e, e' \in E(G), e$ şi e' adiacente în $G\}$.
- ***graful total*** al grafului ***G*** este graful $T(G)$ cu $V(T(G)) = V(G) \cup E(G)$ şi $E(T(G)) = \{xy \mid x, y \in V(G) \cup E(G), x$ şi y adiacente sau incidente în $G\}$.
- ***graful obţinut din G prin inserţia unui vârf*** (z) ***pe o muchie*** (e = vw) este graful $G' = (V(G) \cup \{z\}, E(G) \setminus \{vw\} \cup \{vz, zw\})$ ($z \notin V(G)$, $e \in E(G)$). Două grafuri obţinute prin inserţii succesive de vârfuri pe muchiile aceluiaşi graf se numesc ***homeomorfe***.
- ***graful obţinut din G prin contracţia muchiei*** $e = vw \in E(G)$ este graful $G \mid e = (V(G) \setminus \{v, w\} \cup \{z\}, E([V(G) \setminus \{v, w\}]_G) \cup \{yz \mid yv$ sau $yw \in E(G)\})$. Dacă ***H*** se poate obţine prin contracţii succesive de muchii din graful ***G***, se spune că ***G este contractibil la H***.

Fie $G = (V(G), E(G))$ şi $G' = (V(G'), E(G'))$ două grafuri.

- Dacă $V(G) = V(G')$ atunci ***reuniunea*** celor două grafuri şi ***intersecţia*** lor se definesc $G \cup G' = \{V(G), E(G) \cup E(G')), G \cap G' = (V(G), E(G) \cap E(G'))$.

- Dacă $V(G) \cap V(G') = \emptyset$ atunci $G \cup G' = (V(G) \cup V(G'), E(G) \cup E(G'))$ se numeşte ***reuniunea disjunctivă*** a grafurilor G şi G'. Reuniunea disjunctivă a k grafuri izomorfe cu G se notează kG.
- ***Suma*** a două grafuri G şi G' este graful $G + G' = \overline{\overline{G} \cup \overline{G'}}$.
- ***Produsul cartezian*** al grafurilor G şi G' este graful $G \times G'$ cu $V(G \times G') = V(G) \times V(G')$ şi

$$E(G \times G') = \{(v, w)(v', w') \mid v, v' \in V(G), w, w' \in V(G') \\ v = v' \text{ şi } ww' \in E(G') \text{ sau} \\ w = w' \text{ şi } vv' \in E(G)\}.$$

Clase de grafuri

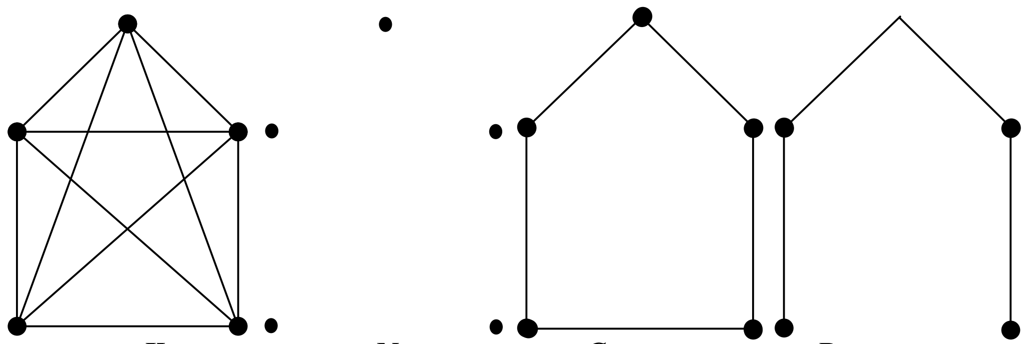
Graful complet de ordin n: K_n cu $|V(K_n)| = n$ şi $E(K_n) = P_2(V(K_n))$.

Graful nul de ordin n: $N_n = \overline{K}_n$.

Circuitul de ordin n: C_n cu $V(C_n) = \{1, \ldots, n\}$ şi $E(C_n) = \{12, 23, \ldots, n-1n, n1\}$.

Drumul de ordin n: $P_n = C_n - e$ ($e \in E(C_n)$).

Exemplu ($n = 5$)



K_5 N_5 C_5 P_5

Un subgraf complet al unui graf G se numeşte ***clică*** a lui G. Cardinalul maxim al unei clici a lui G se numeşte ***numărul de clică*** sau ***numărul de densitate*** al lui G şi se notează $\omega(G)$ (evident $\omega(G) = \alpha(\overline{G})$).

Un ***graf bipartit*** este un graf G cu proprietatea că $V(G)$ se poate partiţiona în două mulţimi independente în G. Dacă S şi T satisfac $S \cup T = V(G)$, $S \cup T = \emptyset$ şi S, T sunt independente şi nevide în G, atunci graful bipartit G se notează $G = (S, T; E(G))$. Remarcăm că $(\forall)$ $e \in E(G)$ are o extremitate în S şi cealaltă în T. Dacă $(\forall)$ $v \in S$ şi $(\forall)$ $w \in T$ $vw \in E(G)$, atunci graful bipartit $G = (S, T; E(G))$ se numeşte ***graf bipartit complet*** şi se notează $K_{s,t}$ unde $s = |S|$ şi $t = |T|$. Exemplu: $K_{3,3}$ este graful

Un graf $G = (V(G), E(G))$ se numeşte ***planar*** dacă poate fi scufundat în plan astfel încât fiecărui vârf să-i corespundă un punct al planului, iar muchiilor le corespund curbe simple ce unesc punctele corespunzătoare vârfurilor lor şi în plus aceste curbe se intersectează (eventual) numai în vârfuri. Un graf care nu-i planar se numeşte ***neplanar***. Exemple minimale de grafuri neplanare sunt grafurile K_5 şi $K_{3,3}$.

Drumuri şi circuite

Fie $G = (V(G), E(G))$ un graf. Se numeşte ***mers de lungime r de la v la w*** în G un şir de vârfuri şi muchii

$$(v =)\ v_0, v_0v_1, \ldots, v_{r-1}, v_{r-1}v_r, v_r\ (= w)$$

v şi w se numesc ***extremităţile mersului***.

Dacă muchiile mersului sunt distincte atunci mersul se numeşte ***parcurs*** în G de la v la w. Dacă vârfurile sunt distincte atunci mersul se numeşte ***drum*** de la v la w.

Dacă $v = w$ atunci mersul (parcursul) se numeşte ***închis***; dacă într-un mers toate vârfurile sunt distincte, cu excepția extremităților, atunci mersul se numeşte ***circuit*** (sau ***drum închis***). Un circuit este ***par*** sau ***impar*** după cum lungimea sa (numărul muchiilor) este pară sau impară. Lungimea celui mai scurt circuit al grafului G (dacă G are circuite) se numeşte ***grația*** grafului G şi se notează cu $g(G)$; lungimea celui mai lung circuit al lui G se numeşte ***circumferința*** lui G şi se notează $c(G)$.

Dacă v şi w sunt vârfuri ale lui G, lungimea celui mai scurt drum de la v la w în G se numeşte ***distanța*** în G de la v la w şi se notează $d_G(v, w)$. ***Diametrul*** grafului G, notat $d(G)$ este $d(G) = \max \{d_G(v, w) \mid v, w \in V(G)\}$.

Definițiile de mai sus se extind, în mod evident, pentru digrafuri singura modificare fiind aceea că se înlocuiesc muchiile cu arce.

Un graf este ***conex*** dacă există un drum între orice două vârfuri ale sale; un graf care nu este conex se numeşte ***neconex***. Orice graf G poate fi unic exprimat ca o reuniune disjunctă de subgrafuri induse conexe şi maximale cu această proprietate; aceste subgrafuri se numesc ***componentele conexe*** ale grafului G (mai precis, se pot defini componentele conexe ca subgrafurile induse de clasele de echivalență determinate pe $V(G')$ de relația de echivalență ρ definită prin $v\rho w$ dacă există în G un drum de la v la w).

Concepte analoage se pot defini şi pentru digrafuri; dacă D este un digraf atunci:

D este ***tare conex*** dacă $\forall\ (v, w) \in V(D) \times V(D)$ există un drum în D de la v la w;

D este ***unilateral conex*** dacă $\forall\ (v, w) \in V(D) \times V(D)$ există în D un drum de la v la w sau de la w la v;

D este ***conex*** dacă $G(D)$, graful suport al lui D, este conex.

Un graf conex care nu are circuite se numeşte ***arbore***. Un graf ale cărui componente conexe sunt arbori se numeşte ***pădure***.

Dacă G este un graf conex, un vârf $v \in V(G)$ cu proprietatea că $G - v$ este neconex se numeşte ***vârf (punct) de articulație***; mai general, o mulțime A de vârfuri ale unui graf G se mai numeşte ***mulțime separatoare de vârfuri (mulțime de articulație)*** dacă $G - A$ este neconex. Fie p un număr întreg pozitiv; un graf G cu măcar p vârfuri este ***p – conex*** dacă $G = K_p$ sau are cel puțin $p + 1$ vârfuri şi nu admite mulțimi separatoare de vârfuri de cardinal mai mic decât p. Evident, G este **1** – conex dacă şi numai dacă este conex. Un graf **2** – conex se numeşte ***bloc***.

Dacă G este un graf conex, o muchie $e \in E(G)$ cu proprietatea că $G - e$ este neconex se numeşte ***punte*** în graful G; mai general, o mulțime A de muchii ale unui graf G se numeşte ***mulțime separatoare de muchii*** dacă $G - A$ este neconex. Un graf G cu măcar p vârfuri este ***p – muchie – conex*** dacă nu admite mulțimi separatoare de muchii de cardinal mai mic decât p.

Numărul de conexiune al lui $\boldsymbol{G}$, notat $\boldsymbol{k(G)}$, (respectiv, ***numărul de muchie – conexiune***, $\lambda(\boldsymbol{G})$) este cel mai mare număr natural $\boldsymbol{p}$ pentru care $\boldsymbol{G}$ este $\boldsymbol{p}$ – conex ($\boldsymbol{p}$ – muchie – conex).

Un graf (sau digraf) se numeşte ***eulerian*** dacă admite un parcurs închis care foloseşte fiecare muchie a grafului (respectiv, fiecare arc al digrafului).

Un (di)graf $\boldsymbol{G}$ se numeşte ***hamiltonian*** dacă are un circuit care trece prin fiecare vârf.

Matrici asociate.

Dacă $\boldsymbol{G = (\{v_1, \ldots, v_n\}, \{e_1, \ldots, e_m\})}$ este un graf, atunci ***matricea de adiacență*** a grafului $\boldsymbol{G}$ este matricea $\boldsymbol{A = (a_{ij})_{n \times n}}$, unde

$$a_{ij} = \begin{cases} 1 \text{ dacă } v_i \text{ şi } v_j \text{ sunt adiacente} \\ 0 \text{ altminteri.} \end{cases}$$

Matricea de incidență a grafului $\boldsymbol{G}$ este matricea $\mathbf{B = (b_{ij})_{n \times m}}$, unde

$$b_{ij} = \begin{cases} 1 \text{ dacă } v_i \text{ şi } e_j \text{ sunt incidente} \\ 0 \text{ altminteri.} \end{cases}$$

În cazul digrafurilor, se pot asocia similar astfel de matrici, în care, evident se poate indica şi orientarea arcelor, folosind elementele **0**, **1** şi **–1**.

Valorile proprii şi polinomul caracteristic ale matricii de adiacență se numesc ***valorile proprii ale grafului***, respectiv, ***polinomul caracteristic al grafului***.

Structuri de date utilizate în reprezentarea (di)grafurilor.

Fie $\boldsymbol{G = (V, E)}$ un (di)graf cu $\boldsymbol{V = \{1, 2, \ldots, n\}}$ şi $\boldsymbol{|E| = e}$. Cele mai uzuale structuri de date utilizate pentru reprezentarea (di)grafului $\boldsymbol{G}$ sunt

a) matricea de adiacență

Dacă $\boldsymbol{A = (a_{ij})_{n \times n}}$ este matricea de adiacență a lui $\boldsymbol{G}$ ($\boldsymbol{a_{ij} := 1}$ **if** $\boldsymbol{ij \in E}$ este **0**) atunci, reprezentarea acesteia cu ajutorul unui tablou bidimensional va necesita $\boldsymbol{O(n^2)}$ operații pentru orice inițializare, deci orice algoritm, care foloseşte o astfel de reprezentare, are complexitatea $\Omega(\boldsymbol{n^2})$.

b) listele de adiacență

Pentru fiecare vârf $\boldsymbol{v \in V}$ se consideră o listă $\boldsymbol{A(v)}$ a vecinilor săi în $\boldsymbol{G}$. Dacă $\boldsymbol{G}$ este graf, atunci $\boldsymbol{A(v)}$ conține $\boldsymbol{N_G(v) = \{ w \mid w \in V}$ şi $\boldsymbol{vw \in E\}}$ iar dacă $\boldsymbol{G}$ este digraf atunci $\boldsymbol{A(v)}$ conține $\boldsymbol{N_G^+(v) = \{ w \mid w \in V}$ şi $\boldsymbol{vw \in E\}}$. Pentru cazul în care $\boldsymbol{G}$ este graf, fiecare muchie $\boldsymbol{vw \in E}$ va genera două elemente în listele de adiacență, unul în $\boldsymbol{A(v)}$ şi celălalt în $\boldsymbol{A(w)}$. Spațiul total de memorie utilizat va fi de $\boldsymbol{O(n + 2e)}$. Pentru cazul în care $\boldsymbol{G}$ este digraf, spațiul de memorie utilizat este de $\boldsymbol{O(n + e)}$. Listele de adiacență pot fi reprezentate ***b1)*** cu ajutorul tablourilor şi ***b2)*** ca structuri dinamice de date.

b1) Se va considera tabloul $\boldsymbol{T}$: **array [1 . . p, 1 . . 2] of integer** unde $\boldsymbol{p = n + 2e}$ dacă $\boldsymbol{G}$ este graf şi $\boldsymbol{p = n + e}$ dacă $\boldsymbol{G}$ este digraf. Vom inițializa $\boldsymbol{T(i, 1) := i}$ $\boldsymbol{\forall i = \overline{1,n}}$; $\boldsymbol{T(i, 2)}$ **:=** adresa în $\boldsymbol{T}$ (indicele) primului element din $\boldsymbol{A(i)}$. Dacă $\boldsymbol{i \in V}$ este vârf izolat, atunci $\boldsymbol{T(i, 2) = 0}$. Dacă $\boldsymbol{T(i, 2) = j}$ atunci $\boldsymbol{T(j, 1)}$ este primul vârf din $\boldsymbol{A(i)}$ iar $\boldsymbol{T(j, 2)}$ este adresa următorului element. Dacă l este adresa ultimului element w din $A(i)$, atunci $T(l, 1) = w$ şi $T(l, 2) = 0$.

b2) Se folosesc următoarele declaraţii:

```
type   pointer = ^elem;
       elem = record
                vârf : 1 . . n;
                leg : pointer;
              end;
```

şi un tablou care conţine trimiteri la începutul fiecărei liste:

```
var    cap : array [1 . . n] of pointer;
       p : pointer;
```

Graful nul de ordin n se iniţializează astfel:

```
for i := 1 to n do cap[i] := nil;
```

Adăugarea ***muchiei ij*** (deci în cazul grafurilor):

```
begin
    new(p);
    p^.vârf := j;
    p^.leg := cap[i];
    cap[i] := p;
    new(p);
    p^.vârf := i;
    p^.leg := cap[j];
    cap[j] := p;
end;
```

Notăm că în descrierea dată am reprezentat listele $A(i)$ ca stive cu elementele din top la adresa ***cap***$[i]$ ($i \in V$). Aceste liste se pot organiza şi sub forma unor cozi sau liste dublu înlănţuite, sau chiar liste circulare, în funcţie de problema care se rezolvă pe (di)graful în cauză.

Sortare topologică. Dat $G = (V, E)$ un digraf, să se determine o numerotare aciclică a vârfurilor sale.

Vom presupune că $V = \{1, 2, \ldots, n\}$, şi că digraful este reprezentat prin listele sale de adiacenţă $A(v)$, deci cu ajutorul tabloului ***cap***$[v]$ $v \in V$.

Problema cere să determinăm vectorul ***ord***$[v]$ $v \in V$, (***ord***$[v]$ = numărul de ordine al vârfului v) astfel încât

$vw \in E \Rightarrow ord[v] < ord[w]$.

Vom transforma şi listele de adiacenţă astfel încât ele să fie ordonate crescător (în noua numerotare). Rezolvarea problemei este simplă şi se bazează pe următoarea

Lemă. *G admite o numerotare aciclică dacă şi numai dacă nu are circuite.*

Demonstraţie. Este evident că dacă G admite o numerotare aciclică atunci G nu are circuite (dacă $v, v_1, v_2, \ldots, v_k, v_1$ sunt vârfurile unui circuit atunci, cum G are o numerotare aciclică, obţinem $ord[v_1] < ord[v_2] < \ldots < ord[v_k] < ord[v_1]$, contradicţie).

Reciproc, dacă G nu are circuite atunci există un vârf $v_0 \in V$ astfel încât $d_G^-(v_0) = 0$ (altfel, datorită finitudinii digrafului, se poate construi un circuit); punem $ord[v_0] := 1$, considerăm $G := G - v_0$ şi repetăm raţionamentul.

Dimensiunea problemei este $O(n + e)$. Vom construi un algoritm care să rezolve problema în timp $O(n + e)$. Acest lucru este posibil datorită unei utilizări judicioase a structurilor de date. Idea algoritmului:

- determinăm gradele interioare ale vârfurilor;
- vârfurile de grad interior **0** le memorăm într-o stivă S_0;

(a) – considerăm vârful din topul stivei S_0 şi-l numerotăm;
(b) – modificăm gradele interioare ale vârfurilor din lista de adiacenţă a vârfului tocmai numerotat;
(c) – în modificarea anterioară crearea unui vârf de grad interior **0** va implica memorarea lui în stiva S_0;
(d) – reluăm secvenţa (a) – (c) până când stiva devine vidă.

Dacă nu s-au numerotat toate vârfurile rezultă că digraful conţine circuite; în cazul epuizării vârfurilor, s-a rezolvat prima parte a problemei. Va trebui să sortăm listele (stivele) de adiacenţă aşa încât ele să fie ordonate crescător în noua numerotare. Vom folosi un caz foarte simplu al sortării cu găleţi (bucket – sort) sau hash – sort:

- colectăm în găleata $k \in \{1, . . ., n\}$ toate vârfurile j cu proprietatea că $jk \in E$ (j şi k sunt noua numerotare).
- parcurgem găleţile de la cea mai mare la cea mai mică $(n \to 1)$ şi golim găleata curentă i astfel: adăugăm i la stiva vecinilor fiecărui vârf j "cules" din găleata i.

După terminarea acestui pas, în stiva vecinilor fiecărui vârf vom avea vârfurile sortate crescător în noua numerotare. Pentru a se putea folosi noua reprezentare a digrafului va trebui, desigur inversată permutarea ***ord***.

Descrierea algoritmului:

```
type pointer = ^elem;
     elem = record
              vârf: 1 . . n;
              prec: pointer;
              end;
var  cap : array [1 . . n] of pointer; {cap[i] = capul stivei vecinilor vârfului i}
     gal : array [1 . . n] of pointer; {găleata i (capul ei) folosită pentru sortare}
     top0 : pointer; {capul stivei S0}
     p1, p : pointer;
     v, i, j, nr : integer;
     d : array [1 . . n] of integer; {vectorul gradelor interioare ale vârfurilor în
subdigraul curent}
     ord : array [1 . . n] of integer; {ord[v] = numărul de ordine al vârfului v}

begin
     nr := 0; {numărul vârfurilor numerotate}
     top0 := nil;
     for i := 1 to n do
```

```
        begin
                p := cap[i];
                while p <> nil do { p ≠ nil }
                        begin
                                j := p^.vârf;
                                d[j] := d[j] + 1;
                                p := p^.prec
                        end
        end;
for i := 1 to n do
        if d[i] = 0 then
                begin
                        new(p);
                        p^.vârf := i;
                        p^.prec := top0;
                        top0 := p;
                end; {s-a creat stiva S0}
while top0 <> nil do {top0 ≠ nil}
        begin
                v := top0^.vârf;
                top0 := top0^.prec;
                nr := nr + 1;
                ord[v] := nr;
                p := cap[v];
                while p <> nil do {p ≠ nil}
                        begin
                                j := p^.vârf;
                                d[j] := d[j] – 1;
                                if d[j] = 0 then
                                        begin
                                                new(p1);
                                                p1^.vârf := j;
                                                p1^.prec := top0;
                                                top0 := p1;
                                        end;
                                p := p^.prec;
                        end;
        end;
if nr < n then
        write (“digraful are circuite”)
else
        begin {hash sort}
                for i := 1 to n  do
                        gal[i] := nil;
                for i := 1 to n do
```

```
                begin
                    p := cap[i];
                    j := ord[i];
                    while p <> nil do
                        begin
                            k := ord(p^.vârf);
                            new(p1);
                            p1^.vârf := j;
                            p1^.prec := gal[k];
                            gal[k] := p1;
                            p := p^.prec;
                        end;
                end;
            for i := 1 to n  do
                cap[i] := nil; {construim noi liste de adiacență}
            for i := n downto 1 do
                begin
                    p := gal[i];
                    while p <> nil do
                        begin
                            k := p^.vârf;
                            new(p1);
                            p1^.vârf := i;
                            p1^.prec := cap[k];
                            cap[k] := p1;
                            p := p^.prec;
                        end;
                end;
            {inversare lui ord}
        end;
end.
```

Descrierea acestui algoritm este tipică pentru anumite libertăți Pascal pe care le vom folosi peste tot în cele ce urmează în scopul măririi lizibilității textului.