Algoritmica grafurilor - Cursul 1

5 octombrie 2018

Cuprins

Pagina cursului şi alte informaţii

Pagina cursului: http://profs.info.uaic.ro/~olariu

Obiective:
Cursurile acoperă noţiunile de bază în Teoria algoritmică a grafurilor. Cunoştinţele acumulate vor fi aplicate în proiectarea unor algoritmi eficienţi pentru rezolvarea problemelor de optimizare combinatorială.

Metode de predare:
Prezentarea video a notelor de curs, care vor fi postate ca fişiere .pdf înaintea fiecărui curs. Aceste fişiere vor conţine şi exerciţiile pentru seminarii.

Pagina cursului şi alte informaţii

Seminarii: E. F. Olariu, A. C. Frăsinaru, A. Policiuc, V. Motroi

Fiecare seminar este dedicat unui număr de exerciţii (postate în avans în notele de curs) cu scopul de a aprofunda conceptele introduse la curs. Studenţii sunt încurajaţi să propună soluţii originale.

Notarea:

- Activitatea de la seminar: prezenţă (max. 8 puncte), participare (max. 10 puncte) - max 18 puncte.
- Teme pentru acasă: trei seturi de exerciţii, max. 14 puncte fiecare - max. 42 puncte
- Examen final scris: max. 60 puncte

Dintr-un maximum de 120 puncte limita de promovare este de 50 puncte.

of G.

Cuprinsul cursului:

- Vocabularul teoriei grafurilor.
- Probleme de drum: parcurgerea sistematică a grafurilor, drumuri de cost minim, conexiune.
- Arbori parţiali de cost minim: union-find, complexitate amortizată.
- Teoria cuplajelor.
- Fluxuri în reţele.
- Reduceri polinomiale între probleme de decizie pe grafuri.
- Abordări ale problemelor **NP**-dificile.
- Grafuri planare.

Aplicaţii ale teoriei grafurilor

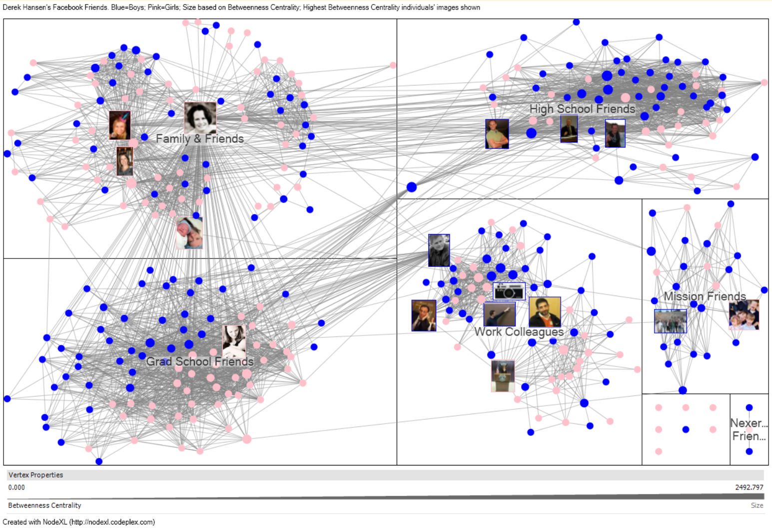


Figure: Facebook/Twitter

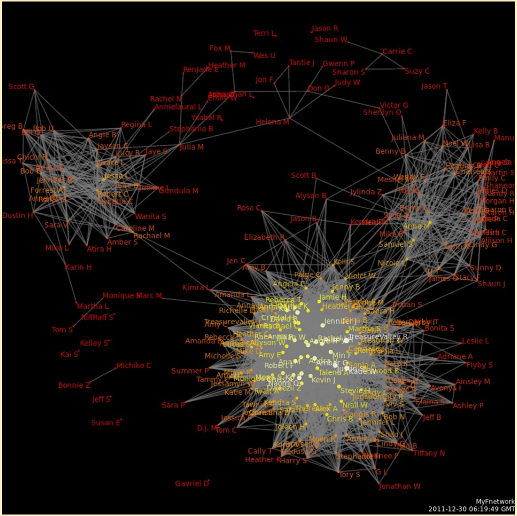


Figure: A Facebook network of a few users

Grafurile sunt folosite pentru analiza reţelelor sociale/de ştiri (precum Facebook sau Twitter) pentru a determina caracteristici cum ar fi:

- nivelul de conexiune, densitatea;
- influenţa utilizatorilor asupra reţelei (centralitatea, potenţialul reţelei sociale (social networking potential);
- nivelul de segmentare: măsura clusterizării;
- robusteţea sau stabilitatea structurală a reţelei.

Analiza reţelelor este folosită pentru

- data mining şi agregare, marketing;
- analiza comportamentului şi predicţie în reţea;
- colectarea de informaţii şi analiza securităţii (supravegherea terorismului şi a crimei organizate) etc.

În afara studiului reţelelor sociale care sunt un subiect la modă astăzi mai sunt multe alte aplicaţii ale teoriei grafurilor:

- **Arbori parţiali de cost minim**: pentru conectarea eficientă a unor puncte de comunicaţie (e. g. în IT&C);
- **Drumuri şi circuite Euleriene**: Problema poştasului chinez - să se determine un drum/circuit de cost minim care trece prin fiecare muchie a unui graf o singură dată (pentru salubrizarea străzilor, expedierea poştei sau a unor servicii, colectarea deşeurilor etc.);
- **Drumuri şi circuite Hamiltoniene**: vizitarea eficientă a unui număr de puncte (dintr-un oraş, dintr-o ţară etc); Problema comis-voiajorului, Problema rutării vehiculelor;

- **Colorarea grafurilor**: colorarea hărţilor (a feţelor unui graf planar), planificarea cursurilor/seminariilor (problema orarului), planificarea unei sesiuni, alocarea frecvenţelor radio mobile, alocarea regiştrilor de memorie.

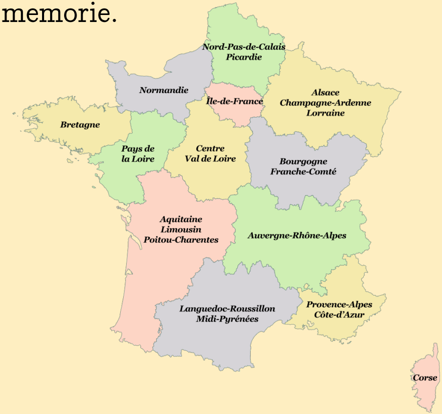


Figure: Regiunile Franţei începând cu 2016

- **Cuplaje**: probleme de asignare, în studii de chimie computaţională şi de chimie matematică asupra materialelor organice.

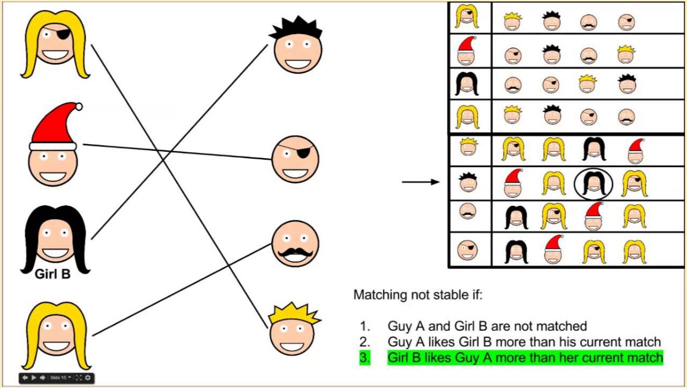


Figure: Gale Shapley algorithm

- **Cuplaje stabile**: repartizarea rezidenţilor în spitale, proceduri de admitere în învăţământul superior, identificarea repartizării optime a organelor donate pentru transplant (e. g. rinichi), problema alocării proiectelor către studenţi etc.
- **Fluxuri de valoare maximă**: determinarea nivelului de încărcare în reţelele de transport pentru îmbunătăţirea condiţiilor de trafic, reconstrucţia imaginilor din proiecţia razelor X (în tomografie), planificarea procesoarelor paralele etc.

Aplicaţii în informatică

- În managementul bazelor de date, bazele de date de tip graf folosesc structurile de date tip graf pentru stocare şi interogare.
- Graph rewriting systems folosite în verificarea sistemelor software.
- Quantum computation.
- Modelarea documentelor web drept grafuri şi clusterizarea acestora.
- Aproximarea şi compresia datelor.
- Modelarea reţelelor de senzori cu grafuri (folosind de exemplu diagrame Voronoi).

Notaţii

Pentru o mulţime finită X:

- $|X| = card(X) \in \mathbb{N}$ este cardinalul lui X;
- Dacă $|X| = k$, atunci X este o k-mulţime;
- $2^X = \mathcal{P}(X)$ este mulţimea părţilor lui X: $2^X = \{Y : Y \subseteq X\}$, $|2^X| = 2^{|X|}$;
- $\binom{X}{k} = \{Y : Y \subseteq X, |Y| = k\}$, $\left|\binom{X}{k}\right| = \binom{|X|}{k}$.

Definiţia 1

Un **graf** *este o pereche* $G = (V, E)$, *unde:*

- $V = V(G)$ *o mulţime finită, nevidă; este mulţimea* **nodurilor** *(***vârfurilor***) lui* G;
- $E = E(G)$ *este o submulţime a lui* $\binom{V(G)}{2}$; *este mulţimea* **muchiilor** *lui* G.

$|G| = |V(G)|$ *este* **ordinul** *grafului* G, *iar* $|E(G)|$ *este* **dimensiunea** *sa.*

Definiţia 2

Dacă $G = (V, E)$ şi $e = uv = vu = \{u, v\} \in E$ este o muchie a lui G, spunem că

- *e* **conectează** *(sau* **leagă***) nodurile u şi v;*
- *nodurile u şi v sunt* **adiacente** *sau u şi v sunt* **vecine***;*
- *e este* **incidentă** *cu u şi v;*
- *u şi v sunt* **capetele** *(***extremităţile***) lui e.*

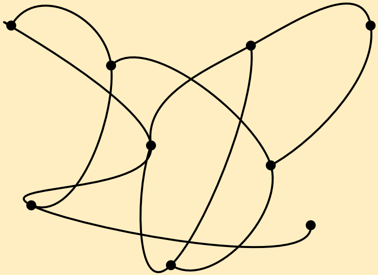
Vecinătatea nodului u este $N_G(u) = \{v \in V(G) : uv \in E(G)\}$.

Două muchii e şi f sunt **adiacente** dacă au un capăt în comun: $|e \cap f| = 1$.

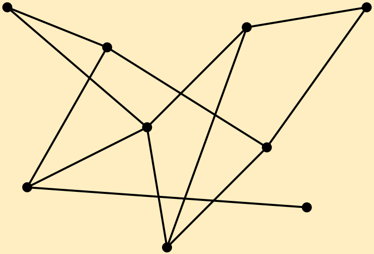
Definiţia grafului

Un graf poate fi reprezentat în plan ca o figură constând dintr-o mulţime de puncte (forme geometrice mici: puncte, cercuri, pătrate etc) corespunzând vârfurilor sale şi curbe care conectează vârfurile corespunzătoare muchiilor din graf.

Un exemplu:



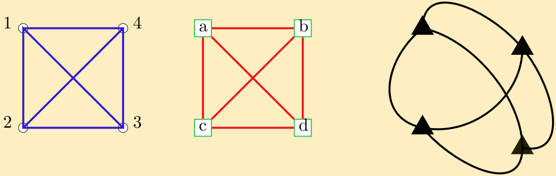
Acelaşi graf din nou:



Putem adăuga etichete (nume, numere etc) şi culori nodurilor şi muchiilor obţinând reprezentări vizuale mai bune.

Mai jos sunt trei reprezentări vizuale ale aceluiaşi graf:

$$G = (\{1, 2, 3, 4\}, \{12, 13, 14, 23, 24, 34\})$$



Definiţia 3

Mulţime stabilă *(sau* **mulţime independentă de noduri***) în* $G = (V, E)$*:* $S \subseteq V$ *astfel încât* $\binom{S}{2} \cap E = \varnothing$.

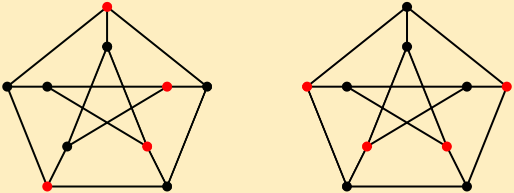
Cu alte cuvinte $S \subseteq V$ este o mulţime stabilă a lui G dacă nu există nicio muchie între nodurile sale.

Notaţie

Cardinalul maxim al unei mulţimi stabile a lui G este **numărul de stabilitate** (sau **numărul de independenţă**) al lui G şi se notează cu $\alpha(G)$.

Exemplu

În următorul graf (al lui Petersen) avem două mulţimi stabile de cardinal maxim (de ce?):



Mulţimi stabile: Problemă de optimizare

P1 Input: G graf.
Output: $\alpha(G)$ şi o mulţime stabilă S cu $|S| = \alpha(G)$.

Mulţimi stabile: Problemă de decizie

SM Instanţă: G graf, $k \in \mathbb{N}$.
Întrebare: Există o mulţime stabilă S în G, astfel încât $|S| \geqslant k$?

NP-completă (Karp, 1972).

Definiţia 4

Cuplaj (**mulţime independentă de muchii**) *în* $G = (V, E)$: $M \subseteq E$ *astfel încât pentru orice* $e, f \in M$, *dacă* $e \neq f$, *atunci* $e \cap f = \varnothing$.

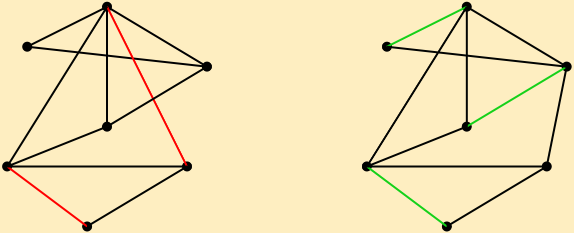
Cu alte cuvinte, $M \subseteq E$ este un cuplaj dacă oricare două muchii ale sale nu noduri în comun.

Notaţie

Cardinalul maxim al unui cuplaj în G este numit **numărul de cuplaj** (**numărul de muchie-independenţă**) al lui G şi se notează cu $\nu(G)$.

Exemplu

Pentr următorul graf sunt marcate două cuplaje cu roşu şi verde (al doilea fiind de cardinal maxim - de ce?):



Cuplaj maxim: **Problemă de optimizare**

P2 Input: G graf.
Output: $\nu(G)$ şi un cuplaj M cu $|M| = \nu(G)$.

Edmonds (1965) a arătat că $P2 \in$ P.

Notă

Problemele $P1$ şi $P2$ sunt similare: în amândouă se cere să se determine un membru de cardinal maxim al unei familii de mulţimi relativ la un graf dat. Ce face diferenţa între ele?

Definiţia 5

*Pentru $p \in \mathbb{N}$, o **p-colorare** a grafului $G = (V, E)$ este o funcţie $c : V \to \{1, \ldots p\}$ astfel încât $c(u) \neq c(v)$ pentru fiecare $uv \in E$.*

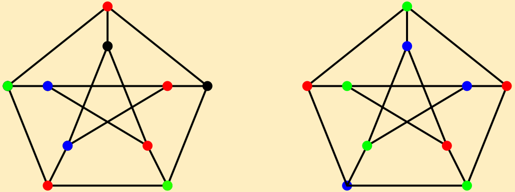
Merită notat că mulţimea tuturor nodurilor cu aceeaşi culoare este o mulţime stabilă (se mai numeşte clasă de colorare). Deoarece noduri adiacente au culori diferite, o p-colorare corespunde unei partiţii a lui V cu cel mult p mulţimi stabile (sau clase de colorare).

Notaţie

Numărul cromatic al grafului G este cea mai mică valoare a lui p astfel încât G are o p-colorare. Acest parametru se notează cu $\chi(G)$. ($\chi(G) \leqslant |G|$ - de ce?)

Exemplu

Pentru următorul graf avem marcate două colorări ($\chi(G) = 3$!)



Colorarea nodurilor: **Problemă de optimizare**

P3 Input: G graf.
Ouput: $\chi(G)$ şi o $\chi(G)$-colorare.

Colorarea nodurilor: **Problemă de decizie**

COL Instanţă: G graf, $p \in \mathbb{N}$.
Întrebare: Există o p-colorare a lui G?

NP-completă pentru $p \geqslant 3$ (Karp, 1972).

Definiţia 6

Pentru $p \in \mathbb{N}$, *a* **p-muchie colorare** *a grafului* $G = (V, E)$ *este o funcţie* $c : E \to \{1, \ldots p\}$ *astfel încât* $c(e) \neq c(f)$ *pentru orice* $e, f \in E$ *cu* $|e \cap f| = 1$.

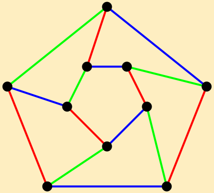
Se poate observa că, dată o p-muchie colorare, o mulţime de muchii cu aceeaşi culoare este un cuplaj. Astfel, o p-muchie colorare corespunde unei partitţii a lui E cu cel mult p cuplaje.

Notaţie

Indexul cromatic al grafului G: cea mai mică valoare a lui p astfel încât G are o p-muchie colorare. Acest parametru este notat cu $\chi'(G)$. ($\chi'(G) \leqslant |E(G)|$ - de ce?)

Exemplu

Pentru următorul graf am marcat o colorare a muchiilor ($\chi'(G) = 3!$)



Colorarea muchiilor: Problemă de optimizare

P3 Input: G graf.
Ouput: $\chi'(G)$ şi o $\chi'(G)$-colorare.

Colorarea muchiilor: Problemă de decizie

COL Instanţă: G graf, $p \in \mathbb{N}$.
Întrebare: Există o p-muchie colorare of G?

NP-completă pentru $p \geqslant 3$ (Holyer, 1984).

Definiţia 7

Două grafuri $G_1 = (V_1, E_1)$ *şi* $G_2 = (V_2, E_2)$ *sunt* **izomorfe** *dacă există o bijecţie* $\varphi : V_1 \to V_2$ *astfel încât pentru orice două noduri* $u_1, v_1 \in V_1$, u_1 *şi* v_1 *sunt adiacente în* G_1 *(i. e.,* $u_1 v_1 \in E_1$*) dacă şi numai dacă* $\varphi(u_1)$ *şi* $\varphi(v_2)$ *sunt adiacente in* G_2 *(i. e.,* $\varphi(u_1)\varphi(u_2) \in E_2$*).*

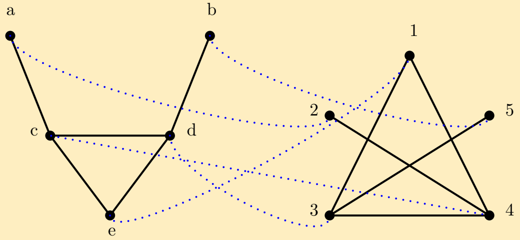
Cu alte cuvinte, două grafuri sunt izomorfe dacă există o bijecţie între mulţimile lor de noduri care induce o bijecţie între mulţimile lor de muchii.

Notaţie

$G_1 \cong G_2$.

Exemplu

Grafurile de mai jos sunt izomorfe.



Testarea izomorfismului: **Problemă de optimizare**

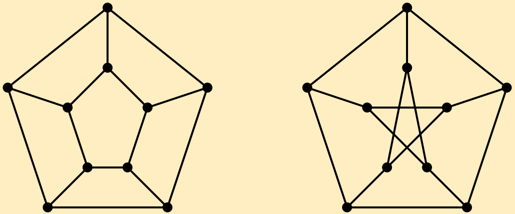
ISO	Input:	grafurile G_1 şi G_2.
	Ouput:	Sunt G_1 şi G_2 izomorfe?

Nu se ştie dacă este în P sau dacă este NP-completă.

Există un algoritm cu timp de execuţie quasipolinomial (i. e., $2^{\mathcal{O}((\log n)^c)}$ pentru un $c > 0$, **Babai, 2015**).

Exemplu

Următoarele două grafuri au aceleaşi ordine, dimensiuni şi secvenţe ale gradelor, dar nu sunt izomorfe (de ce?).



Exerciţiul 1. Pentru $k \in \mathbb{N}^*$, considerăm $G_k = K_2 \times K_2 \times \ldots \times K_2$ (de k ori).

(a) Determinaţi ordinul şi dimensiunea lui G_k

(b) Arătaţi că G_k este bipartit şi determinaţi $\alpha(G_k)$.

Exerciţiul 2.

(a) Există un graf cu gradele $3, 3, 3, 3, 5, 6, 6, 6, 6, 6, 6$?

(b) Există un graf bipartit cu gradele $3, 3, 3, 3, 3, 5, 6, 6, 6, 6, 6, 6, 6, 6$?

(c) Există un graf cu gradele $1, 1, 3, 3, 3, 3, 5, 6, 8, 9$?

Exerciţiul 3.

Doi studenţi, L(azy) şi T(hinky), trebuie să găsească un drum particular între două noduri fixate într-un graf rar G: $|E(G)| = O(|V(G)|)$. L consideră că, deoarece graful este rar, numărul de drumuri dintre cele două noduri trebuie să fie mic, şi o soluţie lazy este să genereze (cu backtracking) toate aceste drumuri şi apoi să reţină drumul dorit. T nu este de acord şi dă următorul exemplu: fie $H = K_2 \times P_{n-1}$ ($n \geqslant 3$); adăugăm la H două noduri noi x şi y fiecare unite prin câte două muchii cu câte una dintre cele două perechi de noduri adiacente de grad 2 din H.

Graful obţinut, G, este rar dar numărul de drumuri dintre x şi y este foarte mare. Explicaţi lui L acest exemplu desenând G, arătând că este rar şi determinând numărul de drumuri dintre x şi y.

Exerciţiul 4. O sesiune de examene trebuie planificată folosind următoarele specificaţii: mulţimea examenelor este cunoscută; fiecare student trimite o listă a examenelor la care doreşte să fie examinat; fiecare examen are loc cu toţi studenţii înscrişi la examen (care este scris); fiecare student poate participa la cel mult un examen într-o aceeaşi zi.

Construiţi un graf cu ajutorul căruia să raspundeţi la următoarele întrebări (prin determinarea unor parametri corespunzători):

(a) Care este numărul maxim de examene care pot fi organizate într-o aceeaşi zi?

(b) Care este numărul minim de zile necesare organizării unei sesiuni?

Exerciţiul 5. (exerciţiul 4 - continuare)

Un programator isteţ şi îndemânatic se întreabă dacă cele două probleme **NP**-hard din exerciţiul anterior nu ar putea fi rezolvate în timp polinomial deoarece graful construit pare să aparţină unei clase speciale de grafuri.

(a) Arătaţi că pentru orice graf dat, G, există un input pentru problema de planificare anterioară astfel încât graful construit (ca mai sus) să fie tocmai G.

Programatorul sugerează următoarea "abordare greedy" pentru a răspunde la a doua întrebare din exerciţiul 4: începând cu prima zi, se planifică zilnic un număr maxim de examene (din mulţimea examenelor neplanificate încă), până când toate examenele vor fi planificate.

b) Arătaţi că această abordare greedy este greşită printr-un contraexemplu.

Exerciţiul 6. Un exemplu de optimizare a unui compilator este tehnica alocării regiştrilor de memorie:cele mai folosite variabile sunt ţinute în registrii cu acces rapid ai procesorului pentru a le avea la îndemână în momentul când compilatorul are nevoie de ele (pentru anumite operaţii CPU).

Construiţi un graf cu ajutorul căruia să raspundeţi la următoarele întrebări (prin determinarea unor parametri corespunzători):

(a) Care este numărul maxim de variabile de care nu este nevoie în acelaşi timp?

(b) Care este numărul minim de regiştri necesari acestor variabile?

Indicaţie: avem două tipuri de obiecte: variabilele (cu valorile lor) şi operaţiile CPU (sau operatorii) care folosesc una sau mai multe variabile.

Exerciţiul 7. (exerciţiul 6 - continuare) Un student se întreabă dacă la întrebările de mai sus (care corespond unor probleme **NP**-hard) nu s-ar putea da un răspuns în timp polinomial deoarece graful construit pare să aparţină unei clase speciale de grafuri.

(a) Arătaţi că pentru orice graf dat, G, există un input pentru problema de alocare a regiştrilor astfel încât graful construit (ca mai sus) să fie tocmai G.

Studentul sugerează următoarea "abordare greedy" pentru a răspunde la a doua întrebare din exerciţiul 6: începând cu primul registru, se alocă fiecărui registru nefolosit un număr maxim de variabile (dintre cele nealocate încă), până când toate variabilele vor fi alocate.

b) Arătaţi că această abordare greedy este greşită printr-un contraexemplu.

Exerciţiul 8. G este numit graf *autocomplementar* dacă G şi complementul său, $\overline{G}$, sunt izomorfe ($G \cong \overline{G}$).

(a) Arătaţi că un graf *autocomplementar* este conex şi că $|G| \equiv 0$ sau 1 *mod* 4.

(b) Determinaţi toate grafurile *autocomplementare* cu cel mult 7 noduri.

(c) Arătaţi că pentru orice graf G există un graf *autocomplementar* H astfel încât G este subgraf indus al lui H.