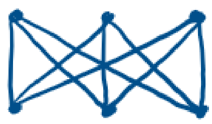
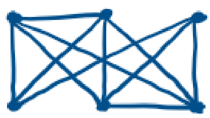
Suport seminar algoritmica grafurilor
IV. Cicluri Euleriene și Hamiltoniene

Probleme:

1. Sunt următoarele grafuri Euleriene?



(a)



(b)

2. Fie un graf complet K_n. Ce valoare are n astfel încât graful conține un ciclu Eulerian? Ce valoare are n astfel încât graful conține un lanț Eulerian? Fie un graf bipartit complet $K_{n,m}$, ce valoare are n și m astfel încât graful conține un ciclu Eulerian?

3. Vârfurile unui graf neorientat sunt numerotate de la $1, \ldots, 100$. Între vârfurile i și j există o muchie dacă $|i - j| \leq 2$. Conține acest graf un lanț Eulerian? Dar un ciclu Eulerian?

4. Să se găsească un ciclu Eulerian în graful din figura 2.

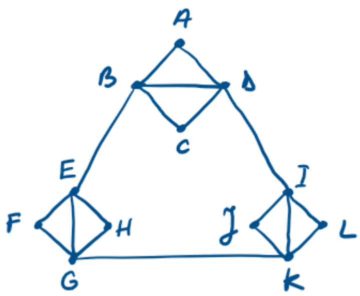


Figura 2

5. Să se găsească un ciclu Eulerian în graful din figura 3 pentru care suma ponderilor este minimă.

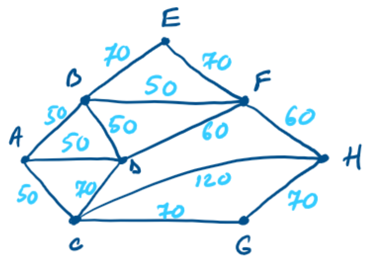


Figura 3

6. Vârfurile unui graf neorientat sunt numerotate de la $1, \ldots, 100$. Între vârfurile i și j există o muchie dacă $|i-j| \leq 2$. Conține acest graf un lanț Hamiltonian? Dar un ciclu Hamiltonian?

7. Să se deseneze un graf 3 regular care să nu conțină un ciclu Hamiltonian.

8. Alin cumpără 100 de mărgele roșii, 150 de mărgele albastre și 200 de mărgele verzi. Să se demonstreze că Alin poate face un colier cu toate mărgelele astfel încât să nu existe mărgele de aceeași culoare vecine în colier.

9. La un hotel se cazează 100 de persoane. La început toți vizitatorii sunt prieteni între ei. În fiecare seară la cină se așează toți vizitatorii la o masă rotundă, din păcate cu ocazia fiecărei mese rotunde persoanele vecine se ceartă între ele și prietenia se termină. Înainte de cină toată lumea se așează la masă astfel încât să nu aibă ca și vecini vizitatori cu care s-au certat la mesele precedente, dacă acest lucru nu este posibil toți pleacă acasă în acea zi. Să se demonstreze că minim 25 de nopți vor fi cazate în hotel cele 100 de persoane.

10. Fie activitățile dintr-un proiect prezentate în tabelul de mai jos împreuna cu durata de execuție a fiecărei activități. Să se determine activitățile critice din proiect. (Dacă apar întârzieri în execuția activităților critice durata proiectului se va extinde peste planificarea inițială.)

Tabela 1: Activităţile, interdependenţă între activităţi şi timpul de execuţie.

activitate	predecesor	durata
A	-	3
B	A	4
C	A	2
D	B	5
E	C	1
F	C	2
G	D, E	4
H	F, G	3

Soluții

Problema 3 Fie un graf simplu, neorientat $G = (V, E)$:

- un lanț eulerian este un lanț simplu care conține toate muchiile din G;
- un ciclu eulerian este un lanț simplu care conține toate muchiile din G și extremitățile lanțului coincid;

- un graf eulerian este un graf simplu care conține un ciclu eulerian.

Graful din cerință este reprezentat în figura 4, se observă că doar două vârfuri au grad impar $\Rightarrow$ graful conține un lanț eulerian dar nu conține un ciclu eulerian.

Corolar 6.3 un graf conex $G = (V, E)$ conține un lanț eulerian dacă și numai dacă are cel mult două vârfuri de grad impar.

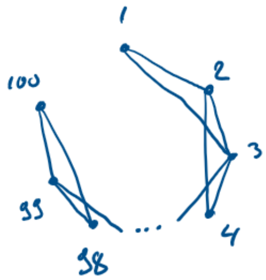


Figura 4

Problema 4 Pentru a rezolva problema se va aplica algoritmul lui Fleury care caută un ciclu eulerian în graf. Algoritmul primește la intrare graful și se presupune că există un ciclu eulerian în graf.

Fleury(G)

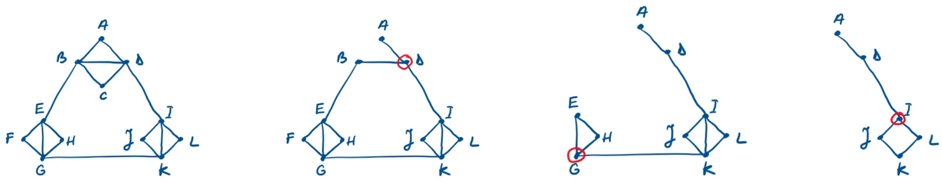
$x = u \in G$
$A = E$
$L = \varnothing$
while $A \neq \varnothing$ **do**
    alege $e \in A$ astfel încât e incidentă lui x ($x \in g(e)$) și dacă se poate e să nu fie punte
    $L = L \cup \{e\}$
    if $|g(e)| = 2$ (dacă s-a parcurs muchia) **then**
        $x = v \in g(e) \setminus \{x\}$ (se merge la extremitatea nevizitată a muchiei e)
    $A = A \setminus \{e\}$
return L

Figura 5 prezintă evoluția grafului dacă se aplică algoritmul lui Fleury pe graful din figura (5a). Se presupune că algoritmul pornește de la vârful A și găsește lanțul $\langle AB, BC, CD \rangle$ (figura (5b)), algoritmul nu alege în continuare muchia AD deoarece aceasta este o muchie punte (dacă se elimină muchia AD din graf graful va avea două componente conexe). Se presupune că lanțul eulerian format până în figura (5c) este format din muchiile $\langle AB, BC, CD, DB, BE, EF, FG \rangle$, în continuare algoritmul nu alege muchia GK deoarece aceasta este o muchie punte (similar cu situația din figura (5d) unde muchia ID este muchie punte și nu poate fi încă adăugată ciclului eulerian). La final ciclu eulerian este $\langle AB, BC, CD, DB, BE, EF, FG, GE, EH, HG, GK, KI, IJ, JK, KL, LI, ID, DA \rangle$.

O altă variantă pentru a găsi un ciclu eulerian într-un graf este algoritmul lui Hierholzer.

Hierholzer

Se identifică un ciclu simplu R_i în G și se marchează muchiile lui R_i.
dacă R_i conține toate muchiile lui G algoritmul se oprește, R_i este eulerian.



(a) (b) (c) (d)

Figura 5: Algoritmul lui Fleury aplicat pe graful de la problema 4.

3: R_i cu conține toate muchile grafului, fie v_i un vârf al lui R_i incident la o muchie nemarcată e_i.
4: Se construiește un ciclu simplu Q_i pornind de la $v_i$din muchiile nemarcate pornind de la e_i. Se marchează muchiile lui Q_i.
5: Se creează R_{i+1} înlănțuind Q_i în R_i la vârful v_i.
6: $i = i + 1$ si se sare la pasul 2.

Problema 5 Problema de a găsi un ciclu Eulerian de pondere minimă mai este cunoscută sub numele de "problema poștașului chinezesc".

Formal problema este definită astfel: pentru un graf conex G să se găsească un ciclu de pondere minimă care conține toate muchiile din G. Problema a apărut ca și răspuns la următoarea întrebare: un poștaș trebuie să își parcurgă ruta asignată înainte de a se întoarce la sediu, poștașul vrea să meargă pe jos cât mai puțin, trebuie găsit traseul de lungime minimă pentru poștaș.

Se consideră graful $G = (V, E)$ cu funcția de pondere asociată $\omega : E \to \mathbb{R}^+$, ponderea unei muchii $w(e)$ poate fi interpretată ca lungimea muchiei (a, b) (a străzii ce trebuie parcursă de poștaș), durata necesară parcurgerii muchiei (a,b), costul necesar parcurgerii muchiei/străzii cu un vehicul, etc. Pentru $H \subseteq G$ și o secvență de muchii $W = e_1, \ldots, e_r$ în G

$$\omega(W) = \sum_{e \in H} \omega(e),\ \omega(W) = \sum_{i=1}^{r} \omega(e_i).$$

Problema poștașului chinezesc cere găsirea unui ciclu W în G de pondere minimă $\omega(W)$ (găsirea unui ciclu eulerian de pondere minimă - poștașul vrea să parcurgă fiecare stradă o singură dată).

Dacă G este eulerian se poate folosi algoritmul lui Fleury sau Hierholzer pentru a găsi un ciclu eulerian în G.

Dacă G nu este eulerian cum se poate găsi un ciclu eulerian de lungime minimă? Se dublează muchii în G astfel încât lungimea ciclului eulerian rezultat să fie minimă. Poștașul va trebui să traverseze de două ori aceeași strada, stradă de lungime (timp, cost) minimă astfel încât traseul său să aibă lungimea minimă.

Un algoritm pentru problema poștașului chinezesc:

poștaș_chinezesc(G)
1: Se identifică vârfurile de grad impar.
2: Se formează toate perechile de vârfuri de grad impar.
3: Pentru fiecare pereche se caută muchiile de cost minim care conectează vârfurile.
4: Se caută perechea pentru care suma ponderilor este minimă.
5: Se dublează muchiile identificate la pasul 4 în graful inițial.
6: Costul drumului poștașului este suma ponderilor muchiilor din graf plus ponderile muchiilor dublate în pasul 5.
7: Se caută un ciclu eulerian în graf.

Algoritmul transformă un graf într-un graf eulerian.

Exemplu Fie graful din figura 6, se identifică vârfurile de grad impar $\{A, B, D, D\}$.

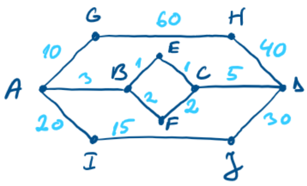


Figura 6

Se formează toate perechile de vârfuri de grad impar $\{\{AB, CD\}, \{AC, BD\}, \{AD, BC\}\}$ și se determină suma ponderilor muchiilor, tabelul 2.

Tabela 2: Perechile de vârfuri de grad impar şi suma ponderilor acestora. Pentru perechea AC lanţul de cost minim care leagă vârfurile A şi C este $\langle AB, BE, EC\rangle$ iar suma ponderilor acestui lanţ este $3 + 1 + 1 = 5$.

AB = 3	AC = 5	AD = 8
CD = 5	BD = 7	BC = 2
$\omega(AB) + \omega(CD) = 8$	$\omega(AC) + \omega(BD) = 12$	$\omega(AD) + \omega(BC) = 10$

Se alege grupul de perechi pentru care suma ponderilor este minimă, în acest caz $\{AB, CD\}$. În graful din figura 6 se dublează muchiile $(A, B), (C, D)$, costul ciclului eulerian pentru poștaș este $\sum_{e\in E} \omega(e) + \omega(AB) + \omega(CD) = 179 + 3 + 5 = 187$.

Problema 6 Fie un graf simplu, neorientat $G = (V, E)$:

- un lanț hamiltonian este un lanț simplu care conține toate vârfurile din G;
- un ciclu hamiltonian este un lanț simplu care conține toate vârfurile din G și extremitățile lanțului coincid;
- un graf hamiltonian este un graf simplu care conține un ciclu hamiltonian.

Un graf cu $n = 5$ vârfuri și muchiile conform cerinței poate fi reprezentat că în figura 7. Se observă că acest graf conține un lanț hamiltonian și un ciclu hamiltonian, un posibil ciclu hamiltonian $\langle 13, 35, 54, 42, 21\rangle$.

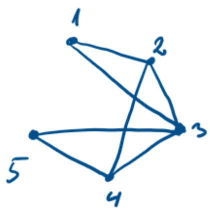


Figura 7

Problema 7 Graful care respectă cerință este reprezentat în figura 8.

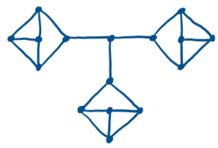


Figura 8

Problema 8 Cu mărgelele cumpărate se poate forma un graf $G = (V, E)$ simplu neorientat și neponderat, $|V| = 450$, în care între două muchii sunt conectate dacă extremitățile au culori diferite.

Pentru graful format gradul vârfurilor este:

$$d(v_{rosii}) = d(v_{verzi}) + d(v_{albastre}) = 350,$$
$$d(v_{verzi}) = d(v_{rosii}) + d(v_{albastre}) = 250,$$
$$d(v_{albastre}) = d(v_{verzi}) + d(v_{rosii}) = 300.$$

Pe baza teoremei lui Dirac graful este hamiltonian ($\delta(G) = 250, \frac{n}{2} = 225, 250 \geq 225$) și se poate construi colierul care să respecte cerința (se poate găsi un ciclu hamiltonian pentru care vârfurile adiacente au culori diferite).

Teorema lui Dirac
Fie G un graf de ordinul $n \geq 3$. Dacă $\delta(G) \geq \frac{n}{2}$ atunci G este hamiltonian.
sau
Fie G un graf de ordinul $n \geq 3$. Dacă $\forall u \in V, d(u) \geq \frac{n}{2}$ atunci G este hamiltonian.

Problema 9 Fie i ziua curentă în care persoanele sunt cazate la hotel, $i = 1, \ldots$. În prima zi toate persoanele cazate sunt prietene, se poate forma un graf complet din care să se aleagă un ciclu hamiltonian care reprezintă schema de așezare a persoanelor la masa. Gradul vârfurilor în prima zi este, în ziua i gradul unui vârf este $99 - 2(i - 1)$.

Pe baza teoremei lui Dirac numărul minim de mese organizate este 25.

Problema 10 Problema se mai numește problema ordonanței sau problema drumului critic și presupune programarea activităților unui proiect. Pentru realizarea unui proiect trebuie realizate un număr mai mare sau mai mic de activități. Deoarece între aceste activități exista o interdependență și o anumită coordonare este foarte importantă programarea desfășurării lor și cunoașterea punctelor sensibile pentru ca obiectivul propus să se realizeze în condiții de eficiență (timp, cost, etc.).

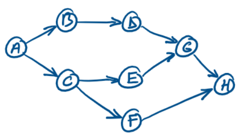
Pentru problema 10 se consideră doar condițiile ce țin de timp, unele activități pot începe mai devreme sau imediat după terminarea altor activități bine precizate. Pentru fiecare activitate se cunoaște durata sa în timp și activitățile ce trebuie realizate înainte ca ea să fie abordată. Drumului critic în graful asociat proiectului permite determinarea timpului minim necesar de realizare a proiectului.

Pentru a determina drumul critic se construiește un graf orientat în care vârfurile reprezinta activitățile proiectului și arcele indică dependență între activități. Figura (9a) prezintă graful asociat proiectului. Pe lângă activitățile descrise în proiect graful va mai conține două vârfuri care reprezintă începutul proiectului și sfârșitul acestuia, figura (9b).

Pentru a începe o activitate trebuie ca toate activitățile care se află pe un drum de la debutul proiectului (vârful start) până la vârful corespunzător activității respective să fie finalizate. Se notează cu *ES* (*Early Start*) timpul cel mai devreme de începere a activității i și se determină ca valoarea maximă a drumurilor de la vârful start la vârful i (suma ponderilor arcelor ce compun drumul este maximă). Timpul cel mai devreme de terminare a activității i, EF (*Early Finish*), se determină ca valoarea maximă a drumurilor de la vârful start la vârful i la care se adaugă timpul de execuție a activității respective ($ES+$ timp_execuție_ activitate_i). După ce se determină momentele cele mai timpurii de începere a activităților se pot determina și momentele cele mai târzii de începere a activităților care nu conduc la o anumită amânare a proiectului. Timpul cel mai târziu de finalizare, LF (*Late Finish*), se determină ca valoarea minimă a drumului de la vârful $final$ la activitatea i. Timpul cel mai târziu de începere a activității, LS (*Late Start*), se determină ca valoarea LF din care se scade timpul de execuție a activității i ($LF-$ timp_execuție_ activitate_i). Pentru vârful *Start* cei patru timpi au valoarea 0 ($ES = EF = LS = LF = 0$), pentru vârful *Final* timpul de execuție este 0.

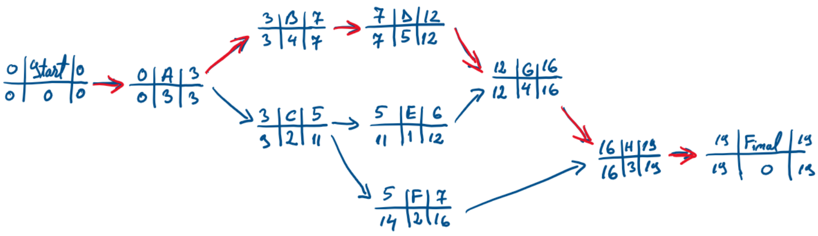
Figura (9b) prezintă vârful *Start* și notația folosită pentru timpii ES, EF, LS, LF, timpul de execuție pentru activitatea *Start* este 0.

Figura (9c) prezintă valoarea timpilor ES, EF, LS, LF pentru fiecare vârf și drumul critic format din vârfurile (activitățile) pentru care acești timpi au aceeași valoare. Drumul critic indică activitățile din proiect care nu trebuie amânate pentru a putea termina proiectul la termen. Activitățile care nu se afla pe drumul critic pot fi pornite în intervalul dat de $[ES, LS]$ și terminate în intervalul $[EF, LF]$ și proiectul se va termina la termen.



(a)

(b)



(c)

Figura 9: Drumul critic pentru proiectul descris în tabelul 1.