INFORMATICĂ · CLASA A XI-A · GRAFURI · LECȚIA 9

Grafuri neorientate (1): noțiuni de bază, grad, reprezentări

Durată: 60 min Nivel: mate-info Înainte: lecțiile x-08

CE VEI ȘTI DUPĂ LECȚIE

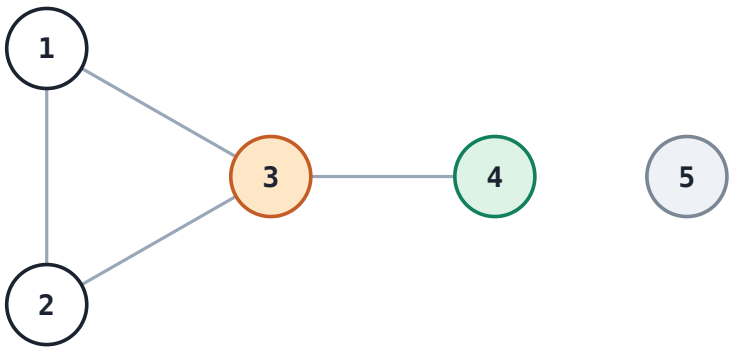
- Cunoști terminologia: vârf (nod), muchie, adiacență, incidență, grad, vârf izolat, vârf terminal.
- Aplici formula sumei gradelor și numărul maxim de muchii.
- Reprezinți un graf prin matrice de adiacență, liste de adiacență și lista muchiilor.
- Recunoști lanțuri, cicluri, subgrafuri și grafuri parțiale.

1. Ce este un graf

Un graf modelează **lucruri legate între ele**: orașe unite prin drumuri, elevi care sunt prieteni, calculatoare dintr-o rețea, pagini web cu legături. Nu contează cum arată desenul, ci **cine cu cine este legat**.

Un **graf neorientat** G = (V, E) are: - **vârfuri** (noduri) V = {1, 2, ..., n}; - **muchii** E: fiecare muchie leagă două vârfuri, [x, y]. Muchia [x, y] este aceeași cu [y, x] (drumul merge în ambele sensuri).

Exemplu: n = 5, muchiile [1,2], [1,3], [2,3], [3,4]. Vârful 5 nu e legat de nimeni.



3: grad 3 (portocaliu) · 4: terminal (verde) · 5: izolat (gri)

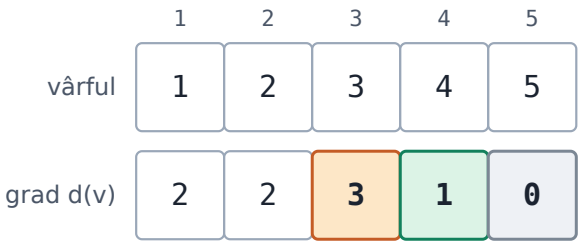
Graful exemplu

Termen	Definiție	În exemplu
vârfuri **adiacente**	legate printr-o muchie	1 și 2
muchie **incidentă** cu x	muchie care îl conține pe x	[3,4] e incidentă cu 3 și cu 4
gradul lui x, d(x)	numărul de muchii incidente cu x	d(3) = 3
vârf **izolat**	grad 0	5
vârf **terminal**	grad 1	4

SFAT

Gradul = „câți vecini are”. Pe desen: numeri liniile care pleacă din vârf.

2. Suma gradelor: formula care apare la fiecare Bac



fiecare muchie se numără de două ori, o dată la fiecare capăt

Gradele

Suma gradelor = 2 · m. De ce: fiecare muchie are două capete, deci contribuie cu 1 la gradul fiecăruia dintre cele două vârfuri, adică 2 în total.

LA BAC

> - Σ d(x) = **2m**; de aici, numărul de vârfuri cu grad **impar** este întotdeauna **par**. - Numărul **maxim** de muchii: **n(n – 1)/2** (fiecare pereche de vârfuri este legată: graf **complet**). - Numărul de grafuri distincte cu n vârfuri **etichetate**: **2^(n(n–1)/2)** (fiecare pereche are muchie sau nu).

Cum se folosesc:

Întrebare	Raționament	Răspuns
Graf cu 7 vârfuri, gradele 1, 2, 2, 3, 3, 4, x. Ce poate fi x?	15 + x trebuie să fie par, deci x impar; x ≤ 6	1, 3 sau 5
Graf complet cu 20 de muchii?	n(n–1)/2 = 20 nu are soluție întreagă (n = 6 dă 15, n = 7 dă 21)	nu există
Un graf are 10 vârfuri și toate au gradul 3. Câte muchii?	10 · 3 / 2	15
Câte grafuri cu 4 vârfuri (etichetate)?	$2^{(4·3/2)} = 2^6$	64

3. Trei moduri de a păstra un graf în memorie

3.1 Matricea de adiacență

`a[i][j] = 1` dacă [i, j] este muchie, altfel 0.

	1	2	3	4	5
1	**0**	1	1	0	0
2	1	**0**	1	0	0
3	**1**	**1**	**0**	**1**	**0**
4	0	0	1	**0**	0
5	0	0	0	0	**0**

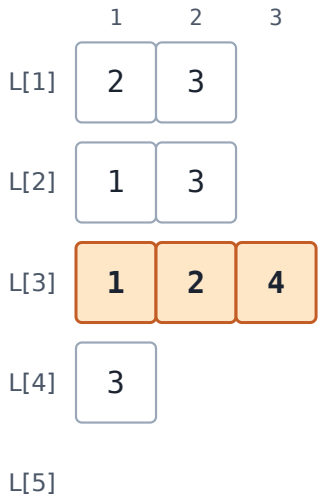
a[i][j] = 1 dacă [i, j] e muchie · simetrică · diagonala 0 · linia 3 însumează gradul lui 3

Matricea

- este **simetrică** față de diagonala principală (muchia e în ambele sensuri);
- diagonala conține doar 0 (nu există muchii de la un vârf la el însuși);
- **gradul** lui i = suma liniei i; numărul de muchii = (numărul de 1) / 2.

3.2 Listele de adiacență

Pentru fiecare vârf, lista vecinilor lui.



gradul lui i = L[i].size() · memorie 2m (nu n²)

Listele

3.3 Lista muchiilor

Un șir de perechi (x, y), exact forma în care se citesc datele: `[1,2] [1,3] [2,3] [3,4]`.

DE ȘTIUT

Care alegi? Matricea: simplă, răspunde în O(1) la „există muchia [i, j]?", dar ocupă n² (bună pentru n ≤ 1000). Listele: ocupă doar 2m, parcurgerea vecinilor unui vârf e rapidă (bune pentru grafuri mari cu puține muchii). La Bac cele mai frecvente sunt matricea și listele.

Citirea grafului: din muchii în matrice

Intrarea: `n m`, apoi `m` perechi `x y`. Programul de mai jos construiește matricea și afișează gradele.

```cpp
#include <iostream>
using namespace std;

int a[101][101], n, m;

int main()
{
    cin >> n >> m;
    for (int k = 1; k <= m; k++)
    {
        int x, y;
        cin >> x >> y;
        a[x][y] = a[y][x] = 1;          // muchia merge în ambele sensuri
    }
    for (int i = 1; i <= n; i++)
    {
        int grad = 0;
        for (int j = 1; j <= n; j++) grad += a[i][j];
        cout << grad << ' ';
    }
    return 0;
}
```

DE CE AȘA?

De ce `a[x][y] = a[y][x] = 1`? Muchia [x, y] este aceeași cu [y, x]. Dacă am nota doar `a[x][y]`, matricea nu ar mai fi simetrică și gradul lui y ar ieși greșit. **De ce suma pe linie?** `a[i][j] = 1` exact pentru vecinii j ai lui i; numărând 1-urile de pe linia i numărăm vecinii.

Trace pentru primele două muchii (`1 2` și `1 3`) și gradul vârfului 3:

Pas	citit	a[x][y]	a[y][x]	grad(3) după parcurgerea liniei 3
1	1 2	a[1][2] = 1	a[2][1] = 1	
2	1 3	a[1][3] = 1	a[3][1] = 1	
3	2 3	a[2][3] = 1	a[3][2] = 1	
4	3 4	a[3][4] = 1	a[4][3] = 1	linia 3 = 1 + 1 + 0 + 1 + 0 = 3

Aceleași date, cu liste de adiacență

```cpp
#include <iostream>
#include <vector>
using namespace std;

vector<int> L[101];
int n, m;

int main()
{
    cin >> n >> m;
    for (int k = 1; k <= m; k++)
    {
        int x, y;
        cin >> x >> y;
        L[x].push_back(y);
        L[y].push_back(x);
    }
    for (int i = 1; i <= n; i++)
    {
        cout << i << ":";
        for (int j = 0; j < (int)L[i].size(); j++) cout << ' ' << L[i][j];
        cout << '\n';
    }
    return 0;
}
```

DE CE AȘA?

De ce `L[i].size()` este gradul? Lista lui i conține exact vecinii lui i. **De ce `push_back` de două ori?** Tot pentru că muchia merge în ambele sensuri: y e vecin al lui x și x e vecin al lui y.

Din matrice: câte muchii, care vârfuri sunt izolate

```cpp
#include <iostream>
using namespace std;

int a[101][101], n;

int main()
{
    cin >> n;
    for (int i = 1; i <= n; i++)
        for (int j = 1; j <= n; j++) cin >> a[i][j];

    int suma = 0;
    for (int i = 1; i <= n; i++)
        for (int j = 1; j <= n; j++) suma += a[i][j];
    cout << suma / 2 << '\n';                       // m = (suma gradelor) / 2

    for (int i = 1; i <= n; i++)
    {
        int grad = 0;
        for (int j = 1; j <= n; j++) grad += a[i][j];
        if (grad == 0) cout << i << ' ';            // vârf izolat
    }
    return 0;
}
```

DE CE AȘA?

De ce împărțim la 2? Fiecare muchie apare de două ori în matrice (în `a[i][j]` și în `a[j][i]`). Suma tuturor valorilor este suma gradelor, deci 2m.

Intrare / ieșire

Intrare	Ieșire

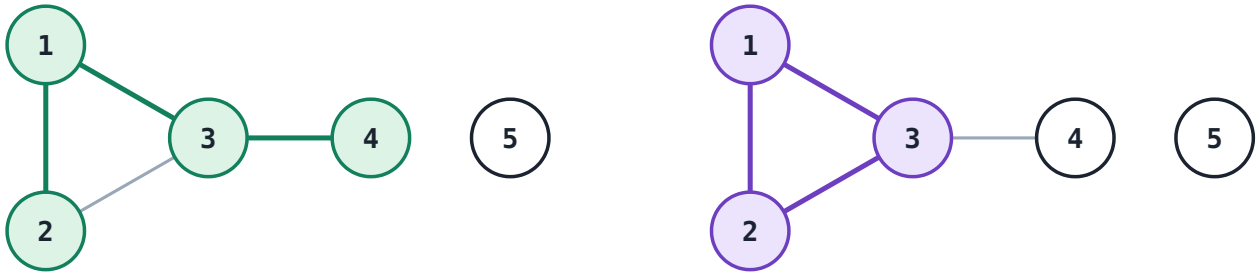
`5 4`
`1 2`
`1 3`
`2 3`
`3 4`
	`2 2 3 1 0` (gradele)

aceeași intrare, listele de adiacență	`1: 2 3`
	`2: 1 3`
	`3: 1 2 4`
	`4: 3`
	`5:`

4. Lanțuri și cicluri

- **lanț**: șir de vârfuri $x_1, x_2, \ldots, x_k$ în care oricare două consecutive sunt adiacente; **lungimea** = numărul de muchii ($k - 1$);
- **lanț elementar**: vârfurile sunt distincte;

- **ciclu**: lanț cu primul vârf = ultimul și muchii distincte; **ciclu elementar**: vârfurile distincte (în afară de primul = ultimul).



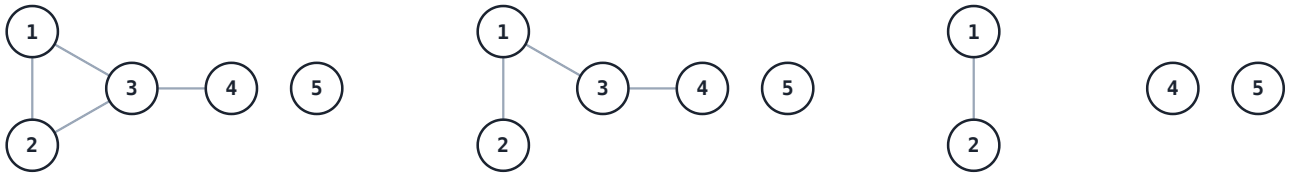
lanț elementar 4, 3, 1, 2 (lungime 3) ciclu elementar 1, 2, 3, 1 (lungime 3)

Lanț și ciclu

ATENȚIE — GREȘELI FRECVENTE

Un ciclu are cel puțin 3 vârfuri distincte: `1, 2, 1` nu e ciclu (folosește aceeași muchie de două ori).

5. Subgraf și graf parțial



G graf parțial: fără [2,3] (aceleași 5 vârfuri) subgraf: fără vârful 3 (și muchiile lui)

Parțial și subgraf

- **graf parțial**: aceleași vârfuri, se elimină unele **muchii**;
- **subgraf**: se elimină unele **vârfuri**, iar odată cu ele dispar toate muchiile incidente.

SFAT

Reține: „**par**țial" = păstrez toate **nodurile** (scot doar muchii). „**sub**graf" = scot noduri. Un graf cu m muchii are 2^m grafuri parțiale (fiecare muchie rămâne sau nu), inclusiv el însuși și graful fără muchii.

6. Greșeli frecvente

Greșeala	Efect	Corect
`a[x][y] = 1;` fără `a[y][x] = 1;`	matrice nesimetrică, grade greșite	ambele
`int a[100][100]` cu n = 100	indici 1..100, depășire	`a[101][101]`

Greșeala	Efect	Corect
gradul = numărul de 1 din toată matricea	număr de 2m, nu d(i)	suma liniei i
m = suma gradelor	de două ori prea mult	m = suma / 2
confuzia parțial / subgraf	răspuns greșit la grilă	parțial = scot muchii
ciclu `1, 2, 1`	nu e ciclu	cel puțin 3 vârfuri

7. La BAC

> **LA BAC**
>
> La Subiectul I apar mai ales: „care este gradul maxim / minim", „câte muchii are graful dat prin matrice/liste", „care vârfuri sunt izolate / terminale", „care dintre șirurile următoare poate fi șirul gradelor" (suma pară, fiecare grad ≤ n − 1), „câte grafuri parțiale / distincte există". Se rezolvă cu formulele din secțiunea 2: scrii gradele, aplici suma = 2m, verifici.

8. Exersează pe hârtie

Răspunsurile sunt la finalul paginii: încearcă întâi singur.

1. Un graf are 6 vârfuri cu gradele 3, 3, 2, 2, 1, x. Câte valori poate lua x (0 ≤ x ≤ 5)? Care?

2. Care este numărul maxim de muchii ale unui graf cu 9 vârfuri? Dar dacă vârful 1 este izolat?

3. Un graf cu 8 vârfuri este regulat de grad 3. Câte muchii are?

4. Câte grafuri parțiale are un graf cu 5 muchii?

5. Desenează graful dat prin matricea de adiacență:

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

Câte muchii are? Care sunt vârfurile terminale?

6. Scrie listele de adiacență pentru graful cu n = 4 și muchiile [1,2], [1,3], [1,4], [3,4].

7. (BAC) Poate fi șirul 4, 3, 3, 2, 2 secvența gradelor unui graf cu 5 vârfuri? Dar 4, 4, 4, 1, 1?

8. Câte grafuri neorientate distincte (cu vârfurile etichetate 1..n) există pentru n = 5?

9. Un graf are 8 vârfuri și 14 muchii, iar fiecare vârf are gradul 3 sau 4. Câte vârfuri au gradul 4?

10. Scrie un program care citește n, m și muchiile și afișează vârfurile de grad maxim.

9. Mai departe (pentru cine vrea mai mult)

Teorema „strângerilor de mână". Numărul de persoane care au dat mâna cu un număr impar de persoane este par: este exact faptul că numărul de vârfuri de grad impar este par.

Secvența gradelor: testul lui Havel-Hakimi. Un șir de grade (sortat descrescător) $d_1 \geq d_2 \geq \ldots$ este valid dacă, scăzând 1 din următoarele d_1 valori și eliminând d_1, șirul rămas este valid. Exemplu: 3, 3, 2, 2, 2 → scoți 3, scazi 1 din următoarele 3: 2, 1, 1, 2 → sortezi 2, 2, 1, 1 → scoți 2: 1, 0, 1 → sortezi 1, 1, 0 → scoți 1: 0, 0 → valid.

Grafuri mari. Pentru n = 100 000 matricea nu încape în memorie (10^{10} elemente); se folosesc doar listele de adiacență.

Multigrafuri și bucle. Dacă două vârfuri au mai multe muchii sau o muchie se întoarce în același vârf, nu mai avem graf „simplu"; matricea ar conține numere > 1 sau valori pe diagonală.

Rezumat

- Graf = vârfuri + muchii; gradul = numărul de vecini; izolat = grad 0; terminal = grad 1.
- Σ grade = 2m; m maxim = n(n−1)/2; grafuri distincte cu n vârfuri: 2^(n(n−1)/2).
- Matricea: simetrică, diagonala 0, gradul = suma liniei. Listele: `L[x].push_back(y)` și `L[y].push_back(x)`.
- Graf parțial = scot muchii; subgraf = scot vârfuri.

Probleme de lucru (pbinfo)

Începe cu acestea, în ordine: sunt alese să urmeze pașii din lecție.

- [ ] #412 **adiacenta** — ușoară
  https://www.pbinfo.ro/probleme/412/adiacenta
- [ ] #414 **listavecini** — ușoară
  https://www.pbinfo.ro/probleme/414/listavecini
- [ ] #416 **grade** — ușoară
  https://www.pbinfo.ro/probleme/416/grade

Temă (pbinfo)

Trimite soluțiile pe pbinfo și notează punctajul obținut. Bifează ce ai terminat.

- [ ] #413 **adiacenta1** — ușoară
  https://www.pbinfo.ro/probleme/413/adiacenta1
- [ ] #417 **gradmax** — ușoară
  https://www.pbinfo.ro/probleme/417/gradmax
- [ ] #430 **izolate** — ușoară
  https://www.pbinfo.ro/probleme/430/izolate
- [ ] #418 **subgraf** — ușoară
  https://www.pbinfo.ro/probleme/418/subgraf

- [ ] #420 **graf-partial** — ușoară
  https://www.pbinfo.ro/probleme/420/graf-partial
- [ ] #474 **veriflant** — ușoară
  https://www.pbinfo.ro/probleme/474/veriflant
- [ ] #2707 **matad** — ușoară
  https://www.pbinfo.ro/probleme/2707/matad

Pentru cei care vor mai mult

Probleme mai grele sau de concurs, pentru cine vrea să meargă dincolo de BAC.

- [ ] #4068 **gradek** — ușoară
  https://www.pbinfo.ro/probleme/4068/gradek

Grile din subiectele de BAC (pbinfo)

Exerciții de tip grilă din variantele de Bacalaureat, rezolvate online pe pbinfo, cu verificare automată a răspunsului.

- [ ] #3 **Exercițiul #3**
  câte grafuri neorientate distincte există cu n vârfuri (2^(n(n-1)/2))
  https://www.pbinfo.ro/?pagina=itemi-evaluare&id=3
- [ ] #67 **Exercițiul #67**
  suma gradelor și numărul de muchii
  https://www.pbinfo.ro/?pagina=itemi-evaluare&id=67
- [ ] #86 **Exercițiul #86**
  graf complet: câte muchii lipsesc / câte are
  https://www.pbinfo.ro/?pagina=itemi-evaluare&id=86
- [ ] #116 **Exercițiul #116**
  graf complet: calcul cu n(n-1)/2
  https://www.pbinfo.ro/?pagina=itemi-evaluare&id=116
- [ ] #119 **Exercițiul #119**
  matricea de adiacență: numărul de valori 1 și numărul de muchii
  https://www.pbinfo.ro/?pagina=itemi-evaluare&id=119
- [ ] #136 **Exercițiul #136**
  valoarea unui grad necunoscut din suma gradelor
  https://www.pbinfo.ro/?pagina=itemi-evaluare&id=136
- [ ] #148 **Exercițiul #148**
  numărul maxim de noduri cu un grad dat
  https://www.pbinfo.ro/?pagina=itemi-evaluare&id=148

- [ ] #159 **Exercițiul #159**
  citirea gradelor dintr-o matrice de adiacență
  https://www.pbinfo.ro/?pagina=itemi-evaluare&id=159

- [ ] #162 **Exercițiul #162**
  liste de adiacență: gradul unui nod
  https://www.pbinfo.ro/?pagina=itemi-evaluare&id=162

- [ ] #1041 **Exercițiul #1041**
  poate fi acest șir secvența gradelor unui graf?
  https://www.pbinfo.ro/?pagina=itemi-evaluare&id=1041

Citește și exersează în plus

- Teorie grafuri neorientate https://www.pbinfo.ro/articole/810/grafuri-neorientate
- Teoria grafurilor — introducere — definiții și istorie (Euler, podurile din Königsberg) https://ro.wikipedia.org/wiki/Teoria_grafurilor
- VisuAlgo — Graph Data Structures — desenezi un graf și vezi matricea, listele și lista de muchii (engleză) https://visualgo.net/en/graphds
- USACO Guide — Introduction to Graphs — aceleași noțiuni, cu probleme de concurs (engleză) https://usaco.guide/silver/intro-graphs

Răspunsuri

Exersează pe hârtie

1. Suma 3+3+2+2+1+x = 11 + x trebuie să fie pară, deci x impar: **x ∈ {1, 3, 5}** (3 valori).

2. Maxim 9·8/2 = **36**. Cu vârful 1 izolat: 8·7/2 = **28**.

3. Suma gradelor 8·3 = 24, deci m = 24/2 = **12**.

4. 2^5 = **32** grafuri parțiale.

5. Muchii: [1,2], [1,4], [2,3] → **3 muchii** (numărul de 1 este 6). Gradele: 2, 2, 1, 1. Vârfuri terminale: **3 și 4**.

6. `L[1]: 2 3 4`, `L[2]: 1`, `L[3]: 1 4`, `L[4]: 1 3`.

7. 4, 3, 3, 2, 2: suma 14 (pară), ținând cont că vârful de grad 4 e legat de toate celelalte, restul rămâne 2, 2, 1, 1 → valid, deci **da**. 4, 4, 4, 1, 1: trei vârfuri de grad 4 sunt legate de toate, deci fiecare din celelalte două are gradul cel puțin 3: **nu**.

8. 2^(5·4/2) = 2^{10} = **1024**.

9. Suma gradelor = 2 · 14 = 28. Dacă toate cele 8 vârfuri ar avea gradul 3, suma ar fi 24; fiecare vârf de grad 4 adaugă 1, deci sunt 28 − 24 = **4** vârfuri de grad 4 (și 4 de grad 3).

10. Calculezi gradele, găsești maximul, apoi afișezi toate vârfurile cu acel grad:

```
int gr[101], mx = 0;
// calculează gr[i] ca suma liniei i (sau L[i].size())
for (int i = 1; i <= n; i++) if (gr[i] > mx) mx = gr[i];
for (int i = 1; i <= n; i++) if (gr[i] == mx) cout << i << ' ';
```