INFORMATICĂ · CLASA A XI-A · RECAPITULARE · LECȚIA 14

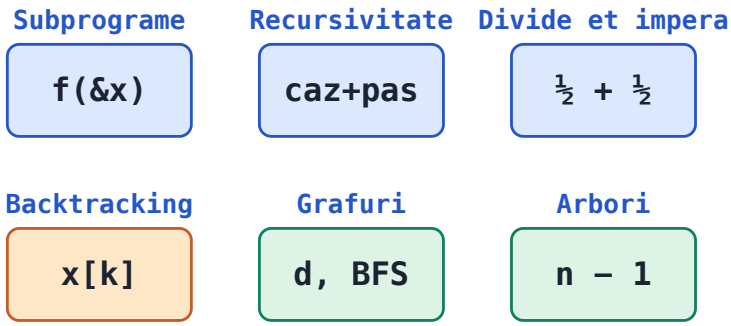
Recapitulare Bac: subprograme, recursivitate, backtracking, grafuri, arbori

Durată: 60 min Nivel: mate-info (Bac) Înainte: lecțiile xi-02, xi-04, xi-08, xi-13

CE VEI ȘTI DUPĂ LECȚIE

- Rezolvi grile de Subiectul I pe toată materia clasei a XI-a.
- Scrii corect, din prima, un subprogram de tip III.1.
- Ai o fișă de formule pentru grafuri și arbori.

1. Harta clasei a XI-a



fiecare temă apare la Subiectul I (grile) și la Subiectele II/III (programare)

Harta

Temă	Ce trebuie să știi pe de rost
subprograme	antet, parametri **valoare** vs **referință** (`&`), valoare returnată vs `void`
recursivitate	caz de bază + pas; ordinea afișării (înainte/după apel); trace pe stivă
divide et impera	împărțire în jumătăți, `T(n) = 2T(n/2) + O(n)` , Hanoi: $2^n - 1$ mutări
backtracking	`x[k]` , `valid(k)` , soluție la `k == n` ; permutări n!, aranjamente, combinări, produs cartezian
grafuri neorientate	Σ grade = 2m; n(n−1)/2; conex ≥ n − 1; (n−1)(n−2)/2 pentru neconex; eulerian = conex + grade pare
grafuri orientate	$\Sigma d^+ = \Sigma d^- = m$; ≤ n(n−1) arce; tare conex ≥ n arce
arbori	n − 1 muchii; vector de tați (rădăcina = 0); nivel, înălțime; frunză = nu apare în `t`

2. Fișa de formule

Noțiune	Formulă / proprietate
graf neorientat, n vârfuri	max n(n−1)/2 muchii; Σ grade = 2m
grafuri neorientate distincte cu n vârfuri	2^(n(n−1)/2)
graf conex	≥ n − 1 muchii
p componente conexe	≥ n − p muchii; p − 1 muchii de adăugat pentru conexitate
neconex, număr maxim de muchii	(n−1)(n−2)/2
graf complet K_n	n(n−1)/2 muchii; fiecare vârf are gradul n − 1
graf parțial	2^m grafuri parțiale (inclusiv el însuși)
eulerian	conex + toate gradele pare
hamiltonian	ciclu prin toate vârfurile; K_n este hamiltonian
bipartit	fără cicluri impare; $K_{p,q}$ are p · q muchii
graf orientat	max n(n−1) arce; Σ d^+ = Σ d^- = m
arbore	n − 1 muchii; ≥ 2 frunze; f = 2 + Σ(d − 2) pe vârfurile cu d ≥ 3
combinatorică	n^k (produs cartezian), A_n^k = n!/(n−k)!, C_n^k = n!/(k!(n−k)!), 2^n submulțimi
Hanoi	2^n − 1 mutări

3. Cum rezolvi o grilă de cod recursiv

Pentru orice grilă cu `f(n)` recursiv: **nu** încerca să „vezi" rezultatul. Scrie un **tabel de apeluri**:

```
void p(int n)
{
    if (n > 0) { p(n - 1); cout << n; p(n - 2); }
}
```

Pentru `p(3)`: apelul `p(3)` face în ordine `p(2)`, afișează 3, `p(1)`.

Apel	Ce face în ordine
p(3)	p(2) → **3** → p(1)
p(2)	p(1) → **2** → p(0)
p(1)	p(0) → **1** → p(-1)
p(0), p(-1)	nu fac nimic

Rezultat: `p(2)` dă `1 2`, deci `p(3)` dă `1 2 3 1`. Pentru `p(4)`: `p(3)` + `4` + `p(2)` = `1 2 3 1 4 1 2` → **1231412**.

4. Grile de antrenament (10 din prima parte)

Fă-le pe hârtie, cronometrează-te (cca. 90 de secunde pe grilă). Răspunsurile cu explicații sunt la final.

G1. Se consideră subprogramul:

```
int f(int n) { if (n < 10) return n; return f(n / 10) + n % 10; }
```

Valoarea lui `f(2025)` este: a) 9 b) 2025 c) 5 d) 7

G2. Ce se afișează la apelul `p(4)` din secțiunea 3? a) 1231412 b) 1213412 c) 12314121 d) 1234123

G3. Utilizând metoda backtracking se generează toate submulțimile cu 3 elemente ale mulțimii {1, 2, 3, 4, 5}, în ordine lexicografică. A cincea soluție generată este: a) {1,3,5} b) {1,4,5} c) {1,3,4} d) {2,3,4}

G4. Un graf neorientat are 8 vârfuri și 4 componente conexe. Numărul maxim de muchii este: a) 10 b) 6 c) 4 d) 12

G5. Un arbore cu rădăcină are vectorul de tați `(4, 4, 0, 3, 3, 1, 1)`. Numărul de frunze este: a) 3 b) 4 c) 2 d) 5

G6. Un graf orientat are 6 vârfuri. Numărul maxim de arce (fără bucle) este: a) 15 b) 30 c) 36 d) 12

G7. Câte cicluri elementare distincte de lungime 3 are graful complet K_4? a) 4 b) 3 c) 12 d) 24

G8. Se consideră:

```
int g(int x) { if (x == 0) return 0; return x % 2 + g(x / 2); }
```

Valoarea lui `g(37)` este: a) 2 b) 3 c) 4 d) 5

G9. Se generează toate numerele cu 3 cifre distincte formate din cifrele 2, 4, 6, în ordine crescătoare. Al patrulea număr generat este: a) 426 b) 462 c) 264 d) 624

G10. Un graf neorientat are 7 vârfuri și muchiile [1,2], [2,3], [4,5], [5,6], [6,7]. Numărul minim de muchii de adăugat pentru a deveni conex este: a) 1 b) 2 c) 0 d) 3

5. Grile de antrenament (10 din partea a doua)

G11. Un graf conex are 10 vârfuri și 14 muchii. Câte muchii trebuie eliminate, cel puțin, pentru a obține un arbore? a) 4 b) 5 c) 6 d) 9

G12. Care este numărul de mutări necesare pentru problema turnurilor din Hanoi cu 6 discuri? a) 31 b) 36 c) 63 d) 64

G13. Arborele cu vectorul de tați `(0, 1, 1, 2, 3, 3, 5)` are înălțimea: a) 2 b) 3 c) 4 d) 7

G14. Se consideră:

```
int h(int a, int b) { if (b == 0) return a; return h(b, a % b); }
```

Valoarea lui `h(48, 18)` este: a) 3 b) 6 c) 12 d) 18

G15. Ce afișează `q(1203)` dacă subprogramul este:

```
void q(int n) { if (n > 0) { cout << n % 10; q(n / 10); } }
```

a) 1203 b) 3021 c) 3012 d) 321

G16. Pe câte moduri se pot așeza 5 elevi pe 3 locuri, ținând cont de ordine? a) 10 b) 15 c) 60 d) 125

G17. Câte șiruri de lungime 4 din literele {a, b, c} nu au două litere consecutive egale? a) 12 b) 24 c) 36 d) 81

G18. Care dintre grafurile următoare este bipartit? a) C_5 (ciclu cu 5 vârfuri) b) C_6 (ciclu cu 6 vârfuri) c) K_3 d) K_4

G19. Un graf neorientat conex are gradele 3, 3, 2, 2, 2, 2. Care afirmație este adevărată? a) este eulerian b) are lanț eulerian, dar nu are ciclu eulerian c) nu are lanț eulerian d) nu există un astfel de graf

G20. Un graf orientat cu 5 vârfuri are 8 arce. Suma gradelor interioare plus suma gradelor exterioare este: a) 8 b) 16 c) 40 d) 20

6. Două subiecte model (III.1 și arbori)

III.1: subprogram

> Scrieți definiția completă a subprogramului `cmax` care primește prin parametrul `n` un număr natural (cu cel mult 9 cifre) și returnează cea mai mare cifră **pară** a lui `n`, sau valoarea −1 dacă `n` nu are cifre pare.

Gândește-te: parcurgi cifrele cu `% 10` și `/ 10`, reții maximul (inițial −1), tratezi doar cifrele pare.

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

int cmax(int n)
{
    int mx = -1;
    do
    {
        int c = n % 10;
        if (c % 2 == 0 && c > mx) mx = c;
        n /= 10;
    } while (n > 0);              // do-while: 0 are o cifră (par)
    return mx;
}

int main()
{
    int n;
    cin >> n;
    cout << cmax(n);
    return 0;
}
```

DE CE AȘA?

De ce `do … while` și nu `while (n > 0)`? Pentru n = 0, `while` nu ar intra deloc și am întoarce −1, deși 0 este cifră pară. `do-while` execută măcar o dată. **De ce `mx = -1` la început?** Este exact valoarea cerută dacă nu găsim nicio cifră pară; orice cifră pară (≥ 0) o va depăși.

Arbori: nodurile de pe ultimul nivel

Se citește un arbore cu rădăcină prin vectorul de tați. Afișați nodurile aflate pe nivelul cel mai mare (cel mai departe de rădăcină), în ordine crescătoare.

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

int t[101], niv[101], n;

int main()
{
    cin >> n;
    for (int i = 1; i <= n; i++) cin >> t[i];
    int mx = 0;
    for (int i = 1; i <= n; i++)
    {
        int y = i;
        while (t[y] != 0) { y = t[y]; niv[i]++; }
        if (niv[i] > mx) mx = niv[i];
    }
    for (int i = 1; i <= n; i++)
        if (niv[i] == mx) cout << i << ' ';
    return 0;
}
```

DE CE AȘA?

De ce două parcurgeri? Mai întâi aflăm nivelul maxim (nu-l știm înainte), apoi afișăm toate nodurile care îl ating. Într-o singură trecere nu am ști ce să afișăm.

7. Greșeli frecvente

ATENȚIE — GREȘELI FRECVENTE

> - Recursivitate fără caz de bază sau cu caz de bază greșit (`n == 0` vs. `n < 10`). - Confuzia „graf parțial” (scot muchii) vs. „subgraf” (scot vârfuri). - Confuzia „hamiltonian” (toate vârfurile) vs. „eulerian” (toate muchiile). - La backtracking: soluțiile sunt generate în ordine lexicografică (în ordinea în care bucla încearcă valorile). - La vectorul de tați: rădăcina este vârful cu valoarea 0, nu vârful 1. - La parametri: fără `&` , modificarea nu se vede în funcția apelantă.

8. Checklist pentru III.1 (subprogram)

- ☐ antetul exact: nume, tip returnat, parametri în ordinea din enunț;

- [ ] parametrii de ieșire prin referință (`&`), dacă rezultatul „se furnizează prin parametru”;
- [ ] nu citesc și nu afișez nimic în subprogram (dacă nu se cere);
- [ ] variabilele locale sunt inițializate;
- [ ] am tratat cazurile speciale din enunț (de exemplu „dacă nu există, se returnează −1”);
- [ ] am verificat pe exemplul din enunț, de mână, inclusiv pe n = 0 și pe un caz limită.

9. La BAC

> **LA BAC**
>
> Subiectul I (30 de puncte) are 6 grile, de obicei: 1 de logică/pseudocod, 1 de subprograme/recursivitate, 1 de backtracking, 2 de grafuri/arbori, 1 de vector de tați sau numărare. Subiectul II are 3 cerințe (o problemă de matrice/șiruri și două pe structuri); Subiectul III: un subprogram (III.1) și un program (III.2), de obicei pe șiruri, matrice sau fișiere. Strategia: începi cu grilele sigure, sari peste cele cu desen mare, revii la final.

10. Exersează pe hârtie

1. Fă un plan de recapitulare: scrie pentru fiecare temă din secțiunea 1 un exemplu mic (cinci rânduri) pe care îl poți rezolva fără carte.

2. Alege două subiecte oficiale de Bac din ultimii ani și rezolvă doar Subiectul I, cronometrat (maxim 30 de minute).

3. Scrie, pe fișa de formule, câte o grilă „tip” pentru fiecare formulă din secțiunea 2 (ex.: „graf cu 8 vârfuri și 5 componente, număr maxim de muchii”).

4. Scrie un program care citește un arbore prin muchii și rădăcină și afișează frunzele (lecția XI.13).

5. Scrie un program care citește un graf și afișează componentele conexe, fiecare pe un rând (lecția XI.11).

Rezumat

- Harta temelor: subprograme, recursivitate, D&I, backtracking, grafuri, arbori.
- Metoda pentru grile de cod: tabel de apeluri sau trace; nu „ghici” rezultatul.
- Formulele de grafuri și arbori se memorează cu exemple mici (n = 4, 5).
- Subprogramele: antet exact, cazuri limită (n = 0), `do-while` când 0 contează.

Probleme de lucru (pbinfo)

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

- [ ] #2838 **nrprime2** — ușoară
  III.1 de Bac.
  https://www.pbinfo.ro/probleme/2838/nrprime2

- [ ] #2811 **triplete** — medie
  III.1 de Bac.
  https://www.pbinfo.ro/probleme/2811/triplete
- [ ] #4060 **gradk** — ușoară
  https://www.pbinfo.ro/probleme/4060/gradk

Temă (pbinfo)

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

- [ ] #2830 **divimpar** — ușoară
  https://www.pbinfo.ro/probleme/2830/divimpar
- [ ] #2836 **fiboimpar** — ușoară
  https://www.pbinfo.ro/probleme/2836/fiboimpar
- [ ] #3104 **b-egal** — ușoară
  https://www.pbinfo.ro/probleme/3104/b-egal
- [ ] #4881 **plus1** — ușoară
  https://www.pbinfo.ro/probleme/4881/plus1
- [ ] #2817 **identice4** — medie
  https://www.pbinfo.ro/probleme/2817/identice4
- [ ] #4071 **ciclul** — ușoară
  https://www.pbinfo.ro/probleme/4071/ciclul

Grile din subiectele de BAC (pbinfo)

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

- [ ] #585 **Exercițiul #585**
  recursivitate: ce se afișează / ce valoare returnează
  https://www.pbinfo.ro/?pagina=itemi-evaluare&id=585
- [ ] #592 **Exercițiul #592**
  recursivitate cu două apeluri
  https://www.pbinfo.ro/?pagina=itemi-evaluare&id=592
- [ ] #469 **Exercițiul #469**
  subprograme: transmiterea parametrilor
  https://www.pbinfo.ro/?pagina=itemi-evaluare&id=469
- [ ] #472 **Exercițiul #472**
  subprograme: valoare returnată
  https://www.pbinfo.ro/?pagina=itemi-evaluare&id=472

- [ ] #1013 **Exercițiul #1013**
  divide et impera: număr de apeluri / ce calculează
  https://www.pbinfo.ro/?pagina=itemi-evaluare&id=1013
- [ ] #380 **Exercițiul #380**
  backtracking: a câta soluție este
  https://www.pbinfo.ro/?pagina=itemi-evaluare&id=380
- [ ] #416 **Exercițiul #416**
  backtracking: câte soluții respectă condiția
  https://www.pbinfo.ro/?pagina=itemi-evaluare&id=416
- [ ] #984 **Exercițiul #984**
  backtracking: combinări
  https://www.pbinfo.ro/?pagina=itemi-evaluare&id=984
- [ ] #73 **Exercițiul #73**
  grafuri: componente conexe maxime
  https://www.pbinfo.ro/?pagina=itemi-evaluare&id=73
- [ ] #116 **Exercițiul #116**
  grafuri: graf complet
  https://www.pbinfo.ro/?pagina=itemi-evaluare&id=116
- [ ] #136 **Exercițiul #136**
  grafuri: gradele unui graf
  https://www.pbinfo.ro/?pagina=itemi-evaluare&id=136
- [ ] #65 **Exercițiul #65**
  arbori: frunze dintr-un vector de tați
  https://www.pbinfo.ro/?pagina=itemi-evaluare&id=65
- [ ] #207 **Exercițiul #207**
  arbori: ce nu poate fi vector de tați
  https://www.pbinfo.ro/?pagina=itemi-evaluare&id=207
- [ ] #1159 **Exercițiul #1159**
  arbori: înălțimea
  https://www.pbinfo.ro/?pagina=itemi-evaluare&id=1159

Citește și exersează în plus

- Subiecte de la examene (Bac) https://www.pbinfo.ro/articole/16/subiecte-de-la-examene
- Subiecte de Bac — informatică (arhiva oficială) — subiectele oficiale, filiera teoretică, mate-info: ia câteva ani la rând și cronometrează-te https://subiecte.edu.ro/
- Informatica pe pbinfo — itemi de evaluare — mii de grile de tip Bac, filtrate pe clasa a XI-a https://www.pbinfo.ro/?pagina=itemi-evaluare-lista

- Visualgo — vezi algoritmii în acțiune — recapitulare vizuală: recursivitate, backtracking, grafuri, arbori (engleză) https://visualgo.net/en

Răspunsuri

Grilele de antrenament

G1. a (9): 2 + 0 + 2 + 5 = 9.

G2. a (1231412): vezi tabelul din secțiunea 3.

G3. a, {1,3,5}: 123, 124, 125, 134, 135.

G4. a (10): fie trei vârfuri izolate și o componentă completă cu 5 vârfuri: 5·4/2 = 10.

G5. b (4): apar ca tați 4, 3, 1; frunze: 2, 5, 6, 7.

G6. b (30): 6 · 5.

G7. a (4): triunghiurile determinate de câte 3 vârfuri: C(4,3) = 4.

G8. b (3): 37 = 100101_2 are trei cifre 1.

G9. b (462): 246, 264, 426, **462**, 624, 642.

G10. a (1): componentele sunt {1,2,3} și {4,5,6,7}; pentru două componente este suficientă o muchie.

G11. b (5): un arbore are 9 muchii, 14 − 9 = 5.

G12. c (63): $2^6 - 1$.

G13. b (3): 7 → 5 → 3 → 1; înălțimea este 3.

G14. b (6): h(48,18) = h(18,12) = h(12,6) = h(6,0) = 6.

G15. b (3021): afișează ultima cifră a lui 1203, apoi pe a lui 120, 12, 1 (de la coadă spre cap).

G16. c (60): A(5,3) = 5·4·3.

G17. b (24): 3 alegeri pentru prima literă, apoi câte 2 pentru fiecare următoare: 3·2·2·2 = 24.

G18. b (C_6): ciclurile de lungime pară se pot colora cu 2 culori; C_5 și K_3, K_4 au cicluri impare.

G19. b: suma gradelor 14 este pară, deci graful există; exact două vârfuri au grad impar, deci are lanț eulerian (între cele două vârfuri de grad 3), dar nu ciclu.

G20. b (16): $\Sigma\, d^+ = \Sigma\, d^- = 8$, deci 8 + 8 = 16.

Subiectele model

Rezolvările sunt în codurile din secțiunea 6 (au fost verificate pe teste, inclusiv n = 0).

Exersează pe hârtie

1–3. Răspuns personal; verifică-te cu fișa de formule și cu lecțiile XI.1–XI.13.

4. Folosește DFS din rădăcină pentru a construi `t[]`, apoi afișează vârfurile care nu apar ca valori în `t` (XI.13, secțiunea 4).

5. O parcurgere (DFS) din fiecare vârf nevizitat; afișezi vârfurile găsite, apoi `'\n'` (XI.11, secțiunea 5).