FIȘĂ DE LUCRU · CLASA A XI-A · FIȘA 8 · ACOPERĂ LECȚIA XI.14

Recapitulare BAC de clasa a XI-a

Durată: 120 min Nivel: mate-info

CE VEI ȘTI DUPĂ LECȚIE

- Urmărești pe hârtie, fără greșeli, subprograme cu parametri prin referință, recursivitate și backtracking.
- Rezolvi itemii de grafuri și arbori din Subiectul I cu formule și cu un desen mic.
- Legi temele între ele: backtracking pe grafuri, recursivitate pe arbori, subprograme de tip III.1.

Ce trebuie să știi (o pagină)

Tip de item	Metoda sigură	Exemplu scurt
parametri valoare / `&`	tabel cu o coloană pe variabilă; `&` = **aceeași căsuță**	`f(x, x)`: o copie și un „alias”
variabilă globală vs locală	locala cu același nume o **ascunde** pe cea globală în acel bloc	`int a = 5;` în `main` nu atinge `a` global
recursivitate	tabel de apeluri: ce face **înainte** de apel, apelul, ce face **după**	ce e după apel se execută la întoarcere, invers
backtracking, „a câta soluție”	numără pe blocuri: câte încep cu 1, cu 2...	n^k, A_n^k, C_n^k, 2^n, $n!$
backtracking, „ce urmează după”	crește ultima poziție care mai poate crește, apoi completează minim	după `1+1+5` (partiții ale lui 7) vine `1+2+2+2`
grafuri	neorientat: Σ grade = 2m, max n(n−1)/2; orientat: Σ d^+ = Σ d^- = m, max n(n−1)	p componente → ≥ n − p muchii
arbore	n − 1 muchii; frunze = vârfurile care **nu apar** în `t`; f = 2 + Σ(d − 2)	rădăcina are `t[r] = 0`
nivel în vectorul de tați	urci din `t` în `t` până la 0, numărând pașii; rădăcina are nivelul **0**	înălțime = nivelul maxim

SFAT

La grilele de Subiectul I, scrie **întâi** tabelul (de apeluri, de variabile, de soluții) și abia apoi alege varianta. Variantele greșite sunt construite exact din greșelile de „văzut cu ochiul”.

Nivelul 1 · Prezice

★ citește și prezice

P1. Doi parametri, aceeași variabilă. Ce afișează secvența? Apoi: dacă și `b` ar fi transmis prin referință (`void h(int &a, int &b) { a += b; b += a; }`), cât ar deveni `x = 2` după `h(x, x)` ?

```
void g(int &a, int b) { a += b; b += a; cout << a << b << ' '; }
int main() { int x = 2; g(x, x); cout << x << ' '; g(x, 5); cout << x; }
```

P2. Ce afișează `p(1234)` ? Dar `p(52816)` ? Spune **în cuvinte** ce face `p` .

```
void p(int n)
{
    if (n == 0) return;
    if (n % 2 == 1) cout << n % 10;
    p(n / 10);
    if (n % 2 == 0) cout << n % 10;
}
```

P3. Partiții. Se generează prin backtracking toate modurile de a scrie 7 ca sumă de numere naturale nenule, cu termenii în ordine crescătoare (nestrict). Primele soluții sunt `1+1+1+1+1+1+1` , `1+1+1+1+1+2` , `1+1+1+1+3` , `1+1+1+2+2` . Ce soluție urmează după `1+1+5` ? Câte soluții sunt în total? Câte au exact 3 termeni?

P4. Vectorul de tați al unui arbore cu 9 noduri este `t = (5, 3, 0, 3, 3, 1, 1, 4, 7)` . Care este rădăcina? Care sunt frunzele? Ce înălțime are? Scrie lanțul elementar de la 9 la 8. Câte noduri are subarborele cu rădăcina 5 (inclusiv 5)?

P5. Graf orientat. n = 6, arcele (1,2), (2,3), (3,1), (3,4), (4,5), (5,4), (6,5). Completează tabelul. Câte componente tare conexe are graful (scrie-le)? În ce vârfuri se poate ajunge din 1? Dar din 6?

vârf	1	2	3	4	5	6
d^+ (ies)						
d^- (intră)						

P6. Global sau local? Ce afișează programul?

```
int a = 1;
int f(int b) { a = a + b; return a * b; }
void afis() { cout << a << ' '; }
int main()
{
    int a = 5;
    int r = f(a);
    cout << r << ' ' << a << ' ';
    afis();
    r = f(r % 7);
    cout << r << ' ';
    afis();
}
```

P7. Ce afișează primele cinci soluții (în ordinea generării) și câte soluții sunt în total pentru apelul `bt(1)` , cu `n = 4` ? Spune în cuvinte ce se generează.

```
void bt(int k)
{
    for (int v = 1; v <= n; v++)
    {
        x[k] = v;
        if (k == 1 || abs(x[k] - x[k - 1]) >= 2)
        {
            if (k == 3) { for (int i = 1; i <= 3; i++) cout << x[i]; cout << ' '; }
            else bt(k + 1);
        }
    }
}
```

P8. Formule, fără desen. Completează.

întrebarea	răspuns
graf neorientat, 10 vârfuri, 3 componente conexe: număr **maxim** de muchii	
același graf: număr **minim** de muchii	
arbore cu 10 noduri: 3 noduri au gradul 3, celelalte au gradul 1 sau 2; câte frunze?	
graf orientat cu 5 vârfuri care **nu** este tare conex: număr maxim de arce	
câte grafuri parțiale cu **exact 2 muchii** are K_4?	

Nivelul 2 · Găsește și repară

★★ găsește și repară

P9. III.1 ratat. Subprogramul trebuie să furnizeze prin `pare` și `impare` numărul de cifre pare, respectiv impare ale lui `n`. Pentru `2735`, în `main` `pare` rămâne cu valoarea pe care o avea înainte de apel, iar pentru `n = 0` se obține `0 0` în loc de `1 0`. Repară cele două greșeli. Colegul tău mai zice că `n % 2` trebuie înlocuit cu `n % 10 % 2`. Are dreptate?

```
void numara(int n, int pare, int &impare)
{
    pare = impare = 0;
    while (n > 0)
    {
        if (n % 2 == 0) pare++; else impare++;
        n /= 10;
    }
}
```

P10. Funcția verifică dacă `v[st..dr]` este palindrom. Pe `(1, 2, 3, 2, 1)` merge, dar pe `(1, 2, 2, 1)` citește din afara vectorului. Scrie apelurile pentru al doilea vector și repară un singur rând.

```
bool pal(int v[], int st, int dr)
{
    if (st == dr) return true;
    if (v[st] != v[dr]) return false;
    return pal(v, st + 1, dr - 1);
}
```

P11. Rânduri amestecate. Monedele sunt `m[] = {1, 2, 5}`. Funcția `bt(k, rest)` afișează toate modurile de a plăti suma, cu monedele în ordine crescătoare (nestrict). Pune rândurile în ordine; unul este în plus. Ce afișează pentru suma 5?

```
A)  x[k] = m[i];
B)  bt(k + 1, rest - m[i]);
C)  if (rest == 0) { afis(k - 1); return; }
D)  rest = rest - m[i];
E)  for (int i = 0; i < 3; i++)
F)  if (m[i] <= rest && (k == 1 || m[i] >= x[k - 1])) {
G)  }
```

P12. Completează golurile. Funcția întoarce numărul de vârfuri din componenta conexă a lui `x` (graf dat prin matricea `a`). Apoi: pentru n = 8 și muchiile [1,2], [2,5], [5,1], [3,4], [6,7], [7,8], [8,6], [6,3], cât returnează `dfs(1)` și câte vârfuri are cea mai mare componentă?

```
int dfs(int x)
{
    viz[x] = 1;
    int c = ______;
    for (int y = 1; y <= n; y++)
        if (a[x][y] == 1 && ______)
            c += ______;
    return c;
}
```

P13. Ce greșeală a făcut elevul? Trei afirmații sunt false și una e adevărată. Pentru cele false dă un contraexemplu.

a. „Un graf neorientat cu n vârfuri și n − 1 muchii este arbore."

b. „Un subprogram `void` nu poate furniza niciun rezultat funcției care îl apelează."

c. „Orice vector de tați al unui arbore cu cel puțin 2 noduri are cel puțin două valori din `1..n` care nu apar în el."

d. „Graful orientat 1→2, 2→3, 3→1 este tare conex; deci și 1→2, 2→3, 1→3 este, fiindcă are aceleași muchii, doar cu altă orientare."

P14. Funcția de mai jos ar trebui să dea nivelul nodului `x` (rădăcina pe nivelul 0). Pentru arborele din P4, programul care o folosește afișează înălțimea 5. Cât este corect? Repară un singur caracter.

```
int niv(int x) { if (t[x] == 0) return 1; return niv(t[x]) + 1; }
```

Nivelul 3 · Scrie cod

★★ scrie cod

Pentru fiecare program, verifică **toate** testele date înainte să spui că e gata.

P15. Ecoul cifrelor impare (III.1). Scrie subprogramul `void transforma(int &n)` care dublează fiecare cifră impară a lui `n` (cel mult 4 cifre), păstrând ordinea. Fără tablouri, fără șiruri de caractere.

Teste: `2735 → 2773355`, `246 → 246`, `1010 → 110110`, `0 → 0`, `9 → 99`.

P16. Jocul 3n + 1. Din `n`: dacă e par, îl înjumătățești; altfel devine `3n + 1`. Scrie o funcție **recursivă** `int pasi(long long n)` care întoarce după câți pași se ajunge la 1. Teste: `6 → 8`, `1 → 0`, `7 → 16`, `27 → 111`.

P17. Echipa de proiect. Se dau `n` elevi cu punctajele lor, `k` și `S`. Afișează (backtracking) toate echipele de `k` elevi, scrise prin indicii lor în ordine crescătoare, câte una pe rând, cu suma punctajelor exact `S`; dacă nu există, `imposibil`. Taie ramurile în care suma depășește deja `S`.

intrare	ieșire
`5 2 10` / `3 7 5 5 8`	`1 2` / `3 4`
`6 3 12` / `2 4 6 3 5 7`	`1 2 3` / `1 4 6` / `2 4 5`
`4 3 100` / `1 2 3 4`	`imposibil`

P18. Harta în trei culori. Se dă un graf (n ≤ 10, apoi m muchii). În câte moduri se pot colora vârfurile cu culorile 1, 2, 3 astfel încât oricare două vârfuri vecine să aibă culori diferite?

Teste: triunghi (`3 3` / `1 2` `2 3` `1 3`) → `6`; lanț (`3 2` / `1 2` `2 3`) → `12`; ciclu (`4 4` / `1 2` `2 3` `3 4` `4 1`) → `18`; K_4 → `0`; `5 0` → `243`.

P19. Strămoșul comun. Se citesc `n`, vectorul de tați și două noduri `x` și `y`. Afișează cel mai apropiat strămoș comun (nodul cel mai de jos care este strămoș și pentru `x`, și pentru `y`; un nod este considerat strămoș al lui însuși). Teste pe arborele din P4: `9 8 → 3`, `6 9 → 1`, `7 9 → 7`, `2 2 → 2`.

P20. Vedeta. Într-o rețea, arcul `x y` înseamnă „x îl urmărește pe y". Vedeta e urmărită de **toți** ceilalți și nu urmărește pe nimeni. Afișează vedeta sau `-1`. Teste: `4 4` / `1 3` `2 3` `4 3` `1 2` → `3`; aceleași arce plus `3 1` → `-1`; `1 0` → `1`.

Nivelul 4 · Pune mintea la contribuție

★★★ pune mintea la contribuție

P21. Fără vecini. Câte submulțimi cu 3 elemente ale lui {1, 2, ..., 7} nu conțin două numere consecutive? Găsește răspunsul fără să le scrii pe toate: dacă `a < b < c`, ce poți spune despre `a`, `b − 1`, `c − 2`? Generalizează pentru `n` și `k` și verifică pentru n = 10, k = 4.

P22. Problema inversă. Un arbore are 8 noduri, dintre care exact 5 sunt frunze. Ce grade pot avea celelalte 3 noduri? Găsește **toate** variantele, apoi scrie un vector de tați pentru una dintre ele.

P23. Ce se întâmplă dacă... în backtracking-ul clasic pentru permutări uiți rândul `uz[v] = false;` de după apelul recursiv? Ce afișează pentru n = 3? Explică de ce, fără să rulezi.

P24. Mini-proiect: Sudoku 4 × 4. Se citesc 4 rânduri de câte 4 cifre (`0` = gol). Completează grila cu 1–4, fără repetiții pe linie, pe coloană și în fiecare pătrat 2 × 2; afișează **prima** soluție găsită sau `imposibil`. Testează pe `1000` / `0021` / `0300` / `4000` (ieșire `1243` / `3421` / `2314` / `4132`) și pe `1100` / `0000` / `0000` / `0000` (`imposibil`). De ce trebuie să verifici și cifrele **date**, înainte de a porni? (Gândește-te la grila plină `1234` / `1234` / `1234` / `1234`.)

P25. Construiește. Există un graf neorientat cu 6 vârfuri și 7 muchii, **neconex**, în care toate vârfurile au grad par? Dacă da, construiește-l și spune ce componente are; dacă nu, explică.

Probleme de lucru în clasă (pbinfo)

Le rezolvăm împreună în ședință, în ordinea de mai jos.

- [ ] `#2838` **nrprime2** — ușoară
  https://www.pbinfo.ro/probleme/2838/nrprime2
- [ ] `#2811` **triplete** — medie
  https://www.pbinfo.ro/probleme/2811/triplete
- [ ] `#4060` **gradk** — ușoară
  https://www.pbinfo.ro/probleme/4060/gradk

Temă pentru acasă (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
- [ ] `#4881` **plus1** — ușoară
  https://www.pbinfo.ro/probleme/4881/plus1
- [ ] `#4071` **ciclul** — ușoară
  https://www.pbinfo.ro/probleme/4071/ciclul