

## Tema 3 - Grafuri și arbori rădăcină

1. **Graf orientat.** Se consideră un graf orientat  $G = (N, A)$  memorat prin liste de adiacență.
  - a. Să se scrie o funcție care verifică pentru orice vârfuri  $x, y \in N$  dacă există drum de la  $x$  la  $y$  și afișează acest drum (dacă există). (1p)
  - b. Să se scrie o funcție care verifică dacă există lanț de la  $x$  la  $y$  și în caz afirmativ afișează un astfel de lanț. (2p).
  - c. Să se scrie o funcție care reținează numărul de componente conexe ale grafului și pentru fiecare dintre acestea afișează nodurile sale. (1p)

Se cere utilizarea unei structuri (sau clase) GRAF.

**Exemplu:** Se consideră graful din figura 1 cu listele de adiacență corespunzătoare.

- a. Între nodurile 0 și 4 există drumul  $\{(0, 2), (2, 7), (7, 4)\}$ . Între nodurile 5 și 1 există drumul  $\{(5, 3), (3, 1)\}$ .
- b. Între nodurile 4 și 1 nu există nici un drum. Între nodurile 4 și 0 există lanț, de exemplu:  $\{(4, 2), (0, 2)\}$ .

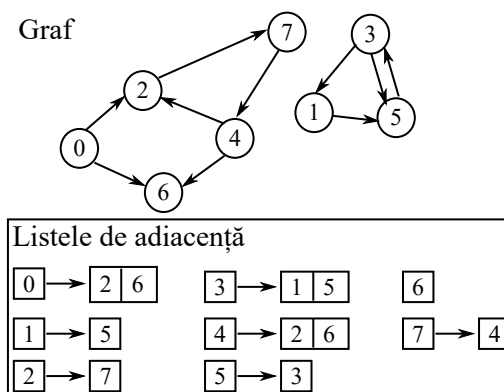


Figure 1: Graf orientat cu reprezentarea cu liste de adiacență

- c. Graful are două componente conexe. Prima conține vârfurile  $\{0, 2, 4, 6, 7\}$ , a doua conține vârfurile  $\{1, 3, 5\}$

2. **Graf neorientat.** Se consideră un graf neorientat conex  $G = (N, A)$  memorat prin liste de adiacență. Pentru un vârf  $v \in N$  vom numi *excentricitate* a lui  $v$  lungimea celui mai scurt drum de la  $v$  la vârful cel mai depărtat de  $v$ . Vom numi *diametru* al grafului excentricitatea maximă a vârfurilor din graf. Se consideră *raza* grafului cea mai mică excentricitate a unui vârf din  $G$ . Să se implementeze funcția excentricitate și să se determine diametrul și raza grafului. Dacă graful nu este conex, se semnalează eroare. (2p)

**Exemplu:** Pentru graful din imaginea 2, excentricitatea nodului 0 este 4, excentricitatea nodului 2 este 3, excentricitatea nodului 5 este 2. Diametrul este 4 și raza este 2.

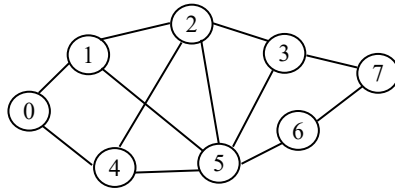


Figure 2: Graf neorientat

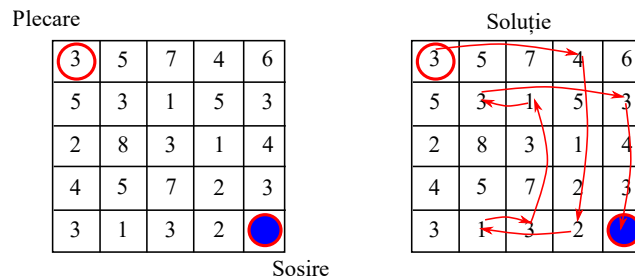


Figure 3: Joc numere exemplu de rezolvare

3. **Joc numere.** Se consideră o matrice  $n \times n$  de numere întregi pozitive (aflate în intervalul  $[0, n - 1]$ ). Un personaj pleacă din colțul stânga-sus și trebuie să ajungă în colțul dreapta jos. La fiecare mutare personajul poate să se deplaseze în sus, în jos, în stânga sau în dreapta cu exact atâtea poziții câte indică numărul din căsuța pe care se află la momentul curent. Nu are voie însă să depășească marginile matricii.

Să se determine numărul minim de mutări necesare pentru a ajunge din colțul stânga sus în colțul dreapta jos. Dacă nu este posibil, să se semnaleze acest

lucru. Un exemplu este prezentat în figura 3. <sup>1</sup>

Variantă: se citește punctul de start și punctul de sosire de la tastatură. (2p)

4. **Arbore sintactic.** Se citește din fișier o expresie aritmetică formată din numere, operatorii de bază (+, -, \*, /) și paranteze. Să se construiască un arbore sintactic corespunzător expresiei. Să se afișeze arborele pe niveluri. **Observație:** într-un astfel de arbore sintactic, care este arbore binar, nodurile interne conțin operatorii aritmetici, iar frunzele conțin operanzii. O operație dintr-un nod intern se efectuează între valorile reprezentate de rezultatele subarborilor copii ai acestui nod. Arborele sintactic permite evaluarea expresiei aritmetice în manieră *bottom-up*. Acest lucru înseamnă că operatorii care se vor aplica mai întâi se află pe niveluri mai mari decât cei care se vor aplica mai apoi. Un exemplu este prezentat în figura 4 a. (2p)

**Exemplu:** Pentru expresia aritmetică  $4+2*((5-1+2)*3+2*(3-1))$  arborele corespunzător este cel din figura 4b. Indicație: se poate folosi forma poloneză postfixată pentru construcția arborelui.

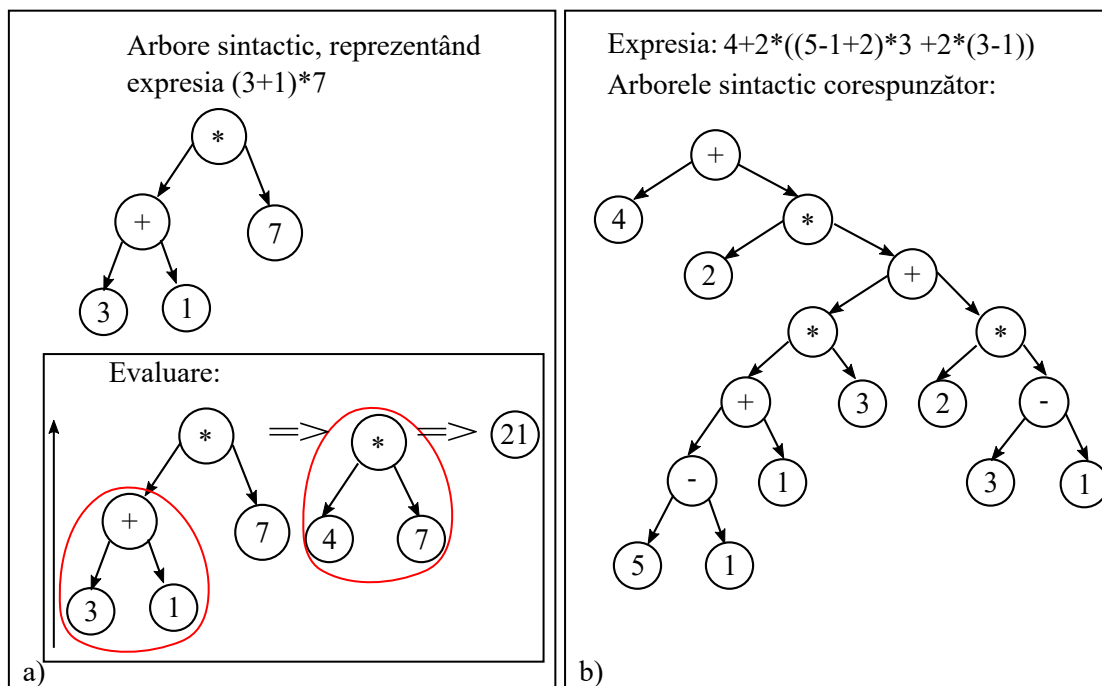


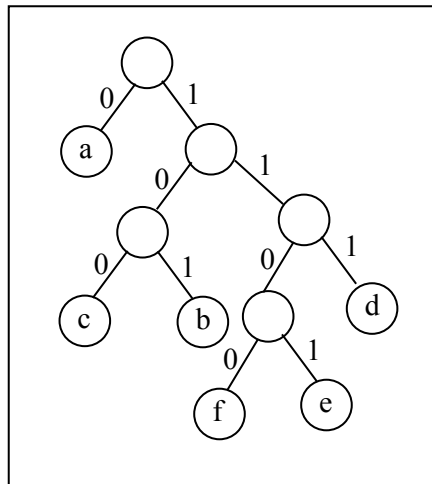
Figure 4: a) Exemplu de arbore sintactic pentru o expresie aritmetică simplă împreună cu modul de evaluare. b) Expresia aritmetică împreună cu arborele sintactic corespunzător.

<sup>1</sup>problemă preluată din "Algorithms" de Jeff Erickson, <http://jeffe.cs.illinois.edu/teaching/algorithms/#book>

5. **Arbore Huffman.** Se citesc dintr-un fișier  $n$  caractere împreună cu codul lor asociat și anume: fiecare caracter va avea asociat un cod format din cifre binare. Codurile vor avea lungime variabilă și niciun cod nu este prefixul altuia. Să se construiască un arbore binar asociat în modul următor. Fiecare caracter va fi inserat într-o frunză a arborelui coborând în arbore de la rădăcină, pentru un caracter 0 pe stânga și pentru un caracter 1 pe dreapta nodului curent. (3p)

**ATENȚIE:** cerința NU este de a determina codurile! (aceasta va fi o problemă la tema 4) ci de a construi arborele, atunci când sunt date codurile!

**Exemplu:** Se consideră perechile caracter - cod asociat:  $a - 0, b - 101, c - 100, d - 111, e - 1101, f - 1100$ . Arborele asociat va fi cel din figură



6. **Refacere arbore din parcurgeri.** Să se refacă un arbore binar cunoscându-se parcurgerile sale în preordine și inordine. Se cere utilizarea unei structuri NOD pentru nodurile arborelui, care să aibă 3 câmpuri: INFO - informație, STANGA - pointer la NOD, DREAPTA - pointer la NOD. Pentru varianta recursivă - 1p. Pentru varianta nerecursivă - 3p.
7. Se citește dintr-un fișier un arbore binar. Să se determine înălțimea acestuia (=lungimea celui mai lung drum de la rădăcină la o frunză). Se cere utilizarea unei structuri NOD pentru nodurile arborelui, care să aibă 3 câmpuri: INFO - informație, STANGA - pointer la NOD, DREAPTA - pointer la NOD, precum și a unei structuri (clase) ARBORE. (1p).
8. **Arbore genealogic.** Se consideră arborele genealogic al unei familii. Construiți o structură corespunzătoare, care să conțină: (2p)

- o funcție de citire a arborelui din fișier
- o funcție de afișare pe niveluri, astfel încât la afișare nivelurile să fie separate prin 2 rânduri libere și pentru fiecare personă să se afișeze numele său, numele părintelui și numărul de copii. (Pentru o afișare grafică a arborelui se oferă suplimentar 2p).
- o funcție, care are ca parametru un nume și afișază numele tatălui și numele copiilor persoanei respective, dacă aceasta se găsește în arbore. Altfel semnalează faptul, că nu a fost găsită persoana. (ce complexitate medie are?)
- o funcție care afișază numărul de generații reprezentate în arbore