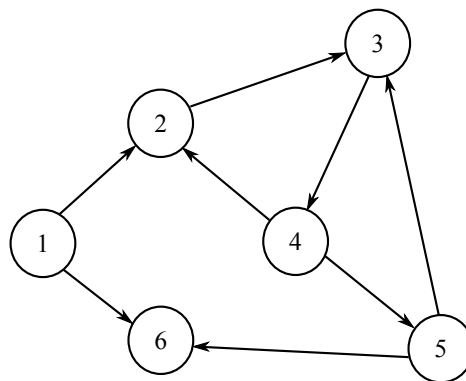


Tema 3 - Grafuri și arbori rădăcină

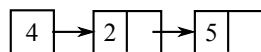
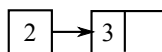
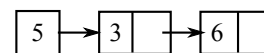
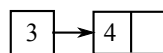
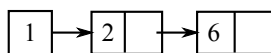
1. Se consideră un graf orientat $G = (N, A)$ memorat prin liste de adiacență. Să se verifice pentru orice vârfuri $x, y \in N$ dacă există drum de la x la y . (2p)
Să se verifice dacă există lanț de la x la y și în caz afirmativ să se afișeze un astfel de lanț.(3p).

Se cere utilizarea unei structuri (sau clase) GRAF.

Exemplu: Se consideră graful din figură:



Listele de adiacență pentru acest graf sunt:



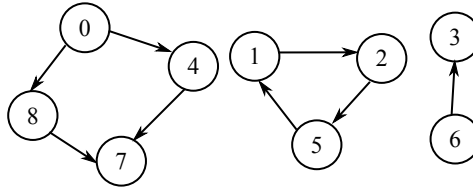
Între nodurile 1 și 5 există drumul $\{(1, 2), (2, 3), (3, 4), (4, 5)\}$.

Între nodurile 5 și 2 există drumul $\{(5, 3), (3, 4), (4, 2)\}$.

Între nodurile 4 și 1 nu există nici un drum. Între nodurile 4 și 1 există lanț: $\{(4, 2), (1, 2)\}$.

2. Se consideră un graf orientat $G = (N, A)$. Să se determine câte componente conexe are și pentru fiecare componentă, care sunt nodurile pe care le conține. Se cere utilizarea unei structuri (sau clase) GRAF. (2p)

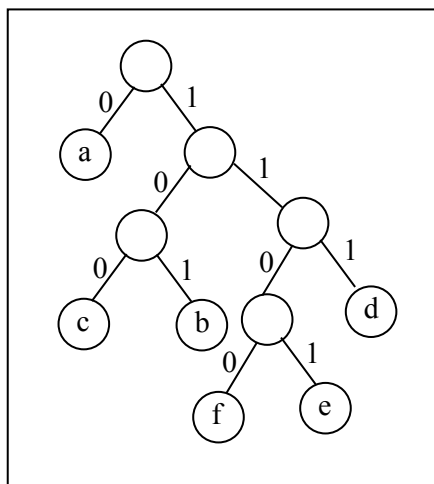
Exemplu: Se consideră graful din figură:



Acest graf are 3 componente conexe. Prima componentă conține nodurile $\{0, 4, 7, 8\}$, a doua componentă conține nodurile $\{1, 2, 5\}$, iar a treia nodurile $\{3, 6\}$.

3. Se consideră un graf orientat $G = (N, A)$ ponderat. Să se determine cu ajutorul algoritmului Floyd-Warshall pentru oricare două noduri, cel mai scurt drum între acestea. În funcția main utilizatorul citește 2 noduri și apoi afișează cel mai scurt drum între ele precum și care este costul drumului. (2p)
4. Se dă un arbore n -ar memorat în reprezentarea fiu-frate. Să se parcurgă în adâncime și în lățime. Se cere utilizarea unei structuri NOD, precum și a unei structuri (clase) ARBORE. (2p)
5. 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. (4p)
- 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. 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).
7. 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. (4p).

Observații: Pentru toate problemele se cere folosirea de structuri: structura graf, structura arbore, etc. Metodele trebuie să facă parte din structurile respective!