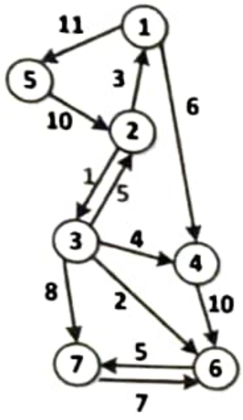
**Nume:**
**Grupa:** **28.01.2025**

## EXAMEN LA ALGORITMI FUNDAMENTALI – VARIANTA 1

**Pentru graful din imaginea din stânga rezolvați cerințele 1-6 și justificați răspunsurile; vecinii unui vârf se consideră în ordine lexicografică**




1) **(0,5p)** Definiți noțiunea de componentă tare conexă. Care sunt componentele tare conexe ale grafului?

2) **(0,5p)** Exemplificați (cu explicații) cum funcționează parcurgerea în adâncime df(3), ilustrând și arborele df asociat și modul în care sunt detectate arcele de întoarcere pe parcursul algoritmului

3) **(0,75p)** Eliminați un număr minim de arce astfel încât graful obținut să admită o sortare topologică și ilustrați un algoritm de sortare topologică pe graful obținut.

4) **(0,75p)** Admite graful un drum eulerian? Dacă nu eliminați un număr minim de arce astfel încât graful format să aibă un drum eulerian, descriind și strategia după care ați ales arcele eliminate. Indicați un drum eulerian în graful inițial/obținut. Enunțați o condiție necesară și suficientă ca un graf orientat să aibă un drum eulerian.

5) **(0,5p)** Descrieți algoritmul Floyd-Warshall pentru determinarea de distanțe într-un graf orientat ponderat cu n vârfuri, detaliind următoarea schemă (se vor respecta numele variabilelor din schemă):

```
Inițializarea matricei D de distanțe cu matricea costurilor
pentru i ← 1, n execută
   pentru j ← 1, n execută
         pentru k ← 1, n execută
               .............
```

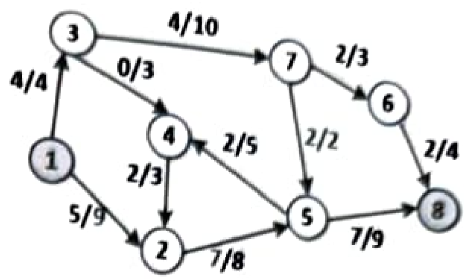
Scrieți ce valori se modifică în matricea D pentru graful din exemplu la etapele i=1, i=2 și i=3 (cu explicații).

6) **(0,5p)** Considerăm graful neorientat H asociat acestui graf obținut astfel: două vârfuri x și y sunt adiacente în H dacă există arc de la x la y sau de la y la x în graf. Costul muchiei de la x la y este minimul dintre costurile arcelor xy și yx - dacă există ambele - sau costul arcului corespunzător xy sau yx, dacă doar unul dintre ele există. Exemplificați pașii algoritmului lui Prim (cu explicații) pornind din vârful 3.

7) **(0,5)** Este corect următorul algoritm de determinare a unui arbore parțial de cost minim al unui graf conex ponderat G = (V, E, w)? Justificați (fără a apela în justificare la modul de funcționare al altor algoritmi; rezultatele folosite trebuie demonstrate și trebuie explicat modul în care se folosesc)

```
T = (V, E = Ø)  -inițial T conține toate vârfurile și nu conține muchii
cat timp T nu e conex executa
      1. Alege o componentă conexă C al lui T cu număr minim de
      vârfuri
      2. Alege o muchie de cost minim e cu o extremitate în C și
      cealaltă nu și adaugă e la T
returneaza T
```

**8)** **(1,25p)** În rețeaua de transport din figura alăturată pe un arc e sunt trecute valorile f(e)/c(e) reprezentând flux/capacitate. Sursa este vârful s=1, iar destinația t=8.




Ilustrați pașii algoritmului Ford-Fulkerson pentru această rețea pornind de la fluxul indicat și alegând la fiecare pas un s-t lanț f-nesaturat de lungime minimă (algoritmul Edmonds-Karp). Indicați o tăietură (s-t tăietură) minimă în rețea (se vor indica vârfurile din bipartiție, arcele directe, arcele inverse și **modul în care tăietura este determinată de algoritm**) și determinați capacitatea acestei tăieturi. Justificați răspunsurile.

**9)** **(1,5)** a) Numărul cromatic al unui graf G este acel p minim cu proprietatea că G este p-colorabil (admite o p-colorare proprie). Dați exemplu de un graf cu numărul cromatic 7. Poate fi un astfel de graf planar? Justificați.

b) Fie M=(V, E, F) o hartă conexă cu n>3 vârfuri cu gradul minim al unui vârf egal cu 2. Arătați că M are cel puțin 3 vârfuri de gradul mai mic sau egal cu 5.

c) Fie G un graf în care toate nodurile au grad par. Arătați că putem orienta arcele din G (adică înlocui muchia xy cu arcul xy sau yx) astfel încât în noul graf orientat obținut orice vârf are gradul intern egal cu cel extern.

**10)** **(0.75p)** Descrieți pe scurt algoritmul de determinare a lungimii maxime a uni subșir comun a două cuvinte (scrieți subproblemele, explicați relațiile de recurență). Ilustrați algoritmul pentru cuvintele "**capace**" si "**paleta**", scriind matricea cu valorile subproblemelor și explicând cum au fost acestea calculate (un subșir al unui cuvânt este format din litere care apar în cuvânt, nu neapărat pe poziții consecutive, dar care respectă ordinea în care apar în cuvânt; de exemplu, pentru cuvintele "**prezenta**" și "**premiant**" cel mai lung subșir comun este "**prent**" ).

**11)** **(1,5p)** Un negustor străbate un ținut deșertic, cu o serie de rute unidirecționate între așezări. El are 2L de apa cu el. Fiecare rută necesită 1L de apa pentru a o străbate. Negustorul vrea să ajungă dintr-o oază inițială în altă oază îndepărtată dată, dar nu poate pleca dintr-un oraș daca nu mai are apă. În deșert unele așezări sunt oaze, acolo negustorul poate să își umple sticla de apa. Descrieți un algoritm de complexitate optimă care determină dacă există un drum pentru negustor și, în caz că există, determină drumul.

**(0,75 soluție corectă + 0,75 discuții complexitate)**