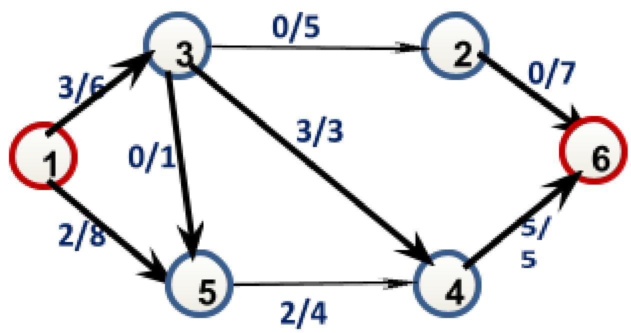
Fluxuri în reţele de transport

1. **Flux maxim**. Se consideră o reţea de transport (care verifică ipotezele din curs) şi un flux în această reţea. Se citesc din fişierul **`retea.in`** următoarele informaţii despre această reţea: numărul de vârfuri n (numerotate 1…n), două vârfuri s şi t reprezentând sursa şi destinaţia, numărul de arce m şi pe câte o linie informaţii despre fiecare arc: extremitatea iniţială, extremitatea finală, capacitatea arcului şi fluxul deja trimis pe arc.
   a) Să se verifice dacă fluxul dat este corect (respectă constrângerile de mărginire şi conservare) şi să se afişeze un mesaj corespunzător. (0.5p)
   b) Să se determine un flux maxim în reţea pornind de la acest flux, prin revizuiri succesive ale fluxului pe s-t lanţuri nesaturate de lungime minimă (Algoritmul Ford - Fulkerson va porni de la fluxul dat, nu de la fluxul vid). Se vor afişa
      - Valoarea fluxului obţinut **şi fluxul pe fiecare arc**
      - Capacitatea minimă a unei tăieturi în reţea şi arcele directe ale unei tăieturi minime **O(mL), L= capacitatea minimă a unei tăieturi** / **O(nm²)** (1.5p)

retea.in	iesire
6	DA
1 6	10
8	1 3 6
1 3 6 3	1 5 4
1 5 8 2	3 2 5
3 2 5 0	3 4 1
3 4 3 3	5 4 4
5 4 4 2	2 6 5
2 6 7 0	4 6 5
4 6 5 5	3 5 0
3 5 1 0	10
	1 3
	5 4



2. **Cuplaj maxim în graf bipartit**. Se citesc din fişierul **`graf.in`** următoarele informaţii despre un graf neorientat **<u>bipartit conex</u>**: numărul de vârfuri n>2, numărul de muchii m şi lista muchiilor (o muchie fiind dată prin extremităţile sale). Să se determine un cuplaj de cardinal maxim în acest graf reducând problema la o problemă de flux maxim şi folosind apoi algoritmul Ford-Fulkerson. Se vor afişa muchiile cuplajului maxim obţinut (vârfurile sunt numerotate 1..n, dar **nu este neapărat ca vârfurile de aceeaşi culoare să fie numerotate consecutiv**) **O(nm)**
   Dacă graful dat la intrare <u>nu</u> este bipartit, se va afișa un mesaj corespunzător și un ciclu impar al grafului. (2p)

graf.in	iesire (nu este unica solutia)
8 9	1 2
1 2	3 4

1 3 2 4 3 4 2 5 3 5 3 7 6 7 7 8	6 7

3. **Construcția unui graf orientat cu secvențele de grade de intrare și ieșire date**. Se citesc din fișierul **`secvente.in`**: un număr natural n>2, o secvență s_1 de n numere naturale și o secvență s_2 de n numere naturale. Să se construiască, dacă se poate, un graf cu secvența gradelor interne s_1 și cu secvența gradelor externe s_2 (reducând problema la o problemă de flux maxim). În caz afirmativ se vor afișa arcele grafului, altfel se va afișa mesajul NU. **O(mn²)** (unde m = suma numerelor din s_1 = numărul de arce ale lui G) (2p)

secvente.in	**iesire (nu este unica solutia)**
3 2 1 1 1 1 2	1 3 2 1 3 1 3 2

4. **LCS**. Se dau două șiruri de caractere, litere mici ale alfabetului englez. Să se afișeze cel mai lung subșir comun al lor. **O(n*m)**. (2p)

lcs.in	**lcs.out**
aaabcd agahbdert	aabd

5. **Ciclu eulerian.** Fiind dat un multigraf *G = (V,E)*, determinati daca acesta este eulerian. In caz afirmativ, gasiti un ciclu eulerian al sau. (1p)

graf.in	**graf.out**
4 6 1 2 1 3 2 2 2 3 3 4 3 4	1 2 2 3 4 3

6. **Ciclu hamiltonian de cost minim.** Fiind dat un graf orientat simplu cu costuri pe arce, să se verifice dacă acesta este hamiltonian. În caz afirmativ, să se determine ciclul hamiltonian de cost minim. (1p)

graf.in	**graf.out**
5 10 0 1 9 0 3 8 1 0 7 1 2 1 1 4 3 2 0 5 2 4 4 3 2 6 4 3 7 4 1 1	26