

Tema obligatorie

Implementați algoritmi eficienți pentru rezolvarea următoarelor probleme. Se vor adăuga comentarii în cod cu ideea algoritmului și complexitatea sa

- **Flux clasic: (0.3 p)**
 - a) <https://www.infoarena.ro/problema/maxflow>
 - b) Modificați algoritmul de la punctul a) astfel încât să afișați tăietura minimă între sursa și orice alt vârf, adică o mulțime de arce cu costul total minim pe care dacă le eliminăm din rețea cel puțin un vârf care era accesibil inițial din sursă după eliminare nu mai este accesibil.
- **Cuplaj maxim implementat cu flux; !!!!alte tipuri de algoritmi de determinare a unui cuplaj maxim nu se punctează: (0.3 p)**
 - <https://www.infoarena.ro/problema/cuplaj>
- **Generare graf orientat cu gradele de intrare și ieșire date: (0.3 p)**
 - <https://www.infoarena.ro/problema/harta>
- **Ciclu Eulerian: (0.3 p)**
 - <https://leetcode.com/problems/valid-arrangement-of-pairs/>
- **Ciclu Hamiltonian: (0.3 p)**
 - <https://leetcode.com/problems/shortest-path-visiting-all-nodes/>
- **Programare dinamica - Șiruri: (0.3 p)**
 - a) <https://leetcode.com/problems/longest-common-subsequence/>
 - b) <https://leetcode.com/problems/shortest-common-supersequence/>

Tema suplimentară

- **Flux maxim de cost minim: (0.2 p)**
 - <https://infoarena.ro/problema/fmcm>
- <https://www.infoarena.ro/problema/drumuri2> (0.4p)
- <https://csacademy.com/contest/archive/task/no-prime-sum/> - cu fluxuri (0.4p)

Indicații:

<https://ali-ibrahim137.github.io/competitive/programming/2020/01/02/maximum-independent-set-in-bipartite-graphs.html>

- <https://www.infoarena.ro/problema/johnie> (0.2p)
- <https://www.infoarena.ro/problema/adn> (0.4p)