

# Tema Nr. 10: Căutare în adâncime

**Timp Alocat:** 2 ore

## Implementare

Se cere implementarea corectă și eficientă a algoritmului de căutare în adâncime (Depth-First Search - DFS) (Capitolul 22.3 din Cormen). Pentru reprezentarea grafurilor va trebui să folosești liste de adiacență. De asemenea va trebui să:

- implementarea algoritmului Tarjan pentru componente tare conexe
- implementezi sortarea topologică (vezi capitolul 22.4)

## Praguri de notare

| Nota | Cerințe  |
|------|--|
| 5    | Implementarea corectă și eficientă la DFS + demo                         |
| 7    | Implementarea corectă și eficientă la sortare topologică + Tarjan + demo |
| 9    | Analiză DFS pe cele 2 cazuri + grafice                                   |
| 10   | Interpretări și discuții   |

## Evaluare

! Înainte de a începe să lucrați la partea de evaluare, asigurați-vă că aveți un algoritm corect!

Demonstrați corectitudinea algoritmului pe un graf de dimensiune mică:

- afișând graful inițial (liste de adiacență)
- componentele puternic conexe ale grafului
- listă de noduri sortate topologic (dacă are / dacă nu are de ce nu are?)

Cum timpul de execuție al algoritmului DFS variază în funcție de numărul de vârfuri ( $|V|$ ) și de numărul de muchii ( $|E|$ ) aveți de făcut următoarele analize:

1. Fixați  $|V|=100$  și variați  $|E|$  între 1000 și 4500 cu un pas de 100. Generați pentru fiecare caz un graf aleator și asigurați-vă că nu generați aceeași muchie de 2 ori. Execută DFS pentru fiecare graf generat și numără operațiile efectuate. Apoi construiește graficul cu variația numărului de operații în funcție de  $|E|$ ;
2. Fixați  $|E|=4500$  și variați  $|V|$  între 100 și 200 cu un pas de 10. Repetă procedura de mai sus și construiește graficul cu variația numărului de operații în funcție de  $|V|$ .

La final interpretați rezultatele.