**Probleme pentru Tehnici Avansate de Programare:**

1. Ordonare clasica O(n^2) (3 algoritmi: [buble sort], [selection sort] si [insertion sort])

<https://github.com/PelleRemus/Facultate/tree/master/Anul%20I/Sem%20I/Sortari/Sortari>.

1. Ordonare eficienta O(n log n) (2 algoritmi: [quick sort] si [merge sort])

<https://github.com/PelleRemus/Facultate/tree/master/Anul%20III/Sem%20II%202019-2020/Tehnici%20avansate%20de%20programare/Sortari%20eficiente/Sortari%20eficiente>.

1. [cautarea binara]

<https://github.com/PelleRemus/Facultate/tree/master/Anul%20I/Sem%20II/Algoritmi%20Fundamentali/BinarySearch/BinarySearch>.

1. Backtracking clasic ([permutari], [aranjamente] si [combinari])

<https://github.com/PelleRemus/Facultate/tree/master/Anul%20III/Sem%20II%202019-2020/Programare%20Paralela/PatratulDiabolic/PatratulDiabolic>.

1. Clasele [stiva], [coada] si [lista ordonata]

<https://github.com/PelleRemus/Facultate/tree/master/Anul%20III/Sem%20II%202019-2020/Tehnici%20avansate%20de%20programare/CoadaListaStiva/CoadaListaStiva>.

1. Clasele ℂ [complex] si ℚ [rational]

<https://github.com/PelleRemus/Facultate/blob/master/Anul%20III/Sem%20II%202019-2020/Programare%20Paralela/NrComplexe/NrComplexe/Program.cs.->complex

<https://github.com/PelleRemus/Facultate/blob/master/Anul%20III/Sem%20II%202019-2020/Programare%20Paralela/ClasaQ/ClasaQ/Program.cs>.-rational

1. Evaluarea expresiilor (Reverse Polish Notation)

<https://github.com/PelleRemus/Facultate/tree/master/Anul%20III/Sem%20II%202019-2020/Tehnici%20avansate%20de%20programare/ReversePolishNotation/ReversePolishNotation>.

1. Algoritmul lui Lee

<https://github.com/PelleRemus/Facultate/tree/master/Anul%20I/Sem%20II/Algoritmi%20Fundamentali/%23Laborator/Drum/Drum>.

1. Grafuri: [parcurgere in latime] si [parcurgere in adancime]

<https://github.com/PelleRemus/Facultate/tree/master/%23Anul%20II/Sem%20I/Met.%20Avansate%20de%20Prog/Grafuri/Grafuri>.

1. Problema teritoriilor si [platouri]

<https://github.com/PelleRemus/Facultate/tree/master/Anul%20I/Sem%20II/Algoritmi%20Fundamentali/%23Laborator/Teritorii/Teritorii> -teritorii

<https://github.com/PelleRemus/Facultate/tree/master/Anul%20I/Sem%20II/Algoritmi%20Fundamentali/%23Laborator/Platouri%20Matrice/Platouri%20Matrice> -platouri

1. Colorarea hartii (grafuri)

<https://github.com/PelleRemus/Facultate/tree/master/%23Anul%20II/Sem%20I/Met.%20Avansate%20de%20Prog/ColorareaHartii/ColorareaHartii>

1. Dijkstra (grafuri)

<https://github.com/PelleRemus/Facultate/tree/master/%23Anul%20II/Sem%20I/Met.%20Avansate%20de%20Prog/GrafDrumCostMinim/GrafDrumCostMinim>.

1. Graf Hamiltonian si Eulerian

<https://github.com/PelleRemus/Facultate/tree/master/Anul%20III/Sem%20II%202019-2020/Tehnici%20avansate%20de%20programare/GrafHamiltonEuler/GrafHamiltonEuler>.

1. Arbore de cost minim Kruskall, Prim

<https://github.com/PelleRemus/Facultate/tree/master/%23Anul%20II/Sem%20I/Met.%20Avansate%20de%20Prog/Kruskall%20Algorythm/Kruskall%20Algorythm>.

La “Prim” cred ca se refera sa verificaream daca la arborele minim, costul este prim, deci codul este foarte putin diferit.

1. Problema Damelor (backtracking)

<https://github.com/PelleRemus/Facultate/tree/master/Anul%20I/Sem%20II/Algoritmi%20Fundamentali/Dame/Dame>.

1. Produsul a doua matrici

<https://github.com/PelleRemus/Facultate/tree/master/Anul%20III/Sem%20II%202019-2020/Tehnici%20avansate%20de%20programare/ProdusMatrici/ProdusMatrici>.

1. Determinantul (recursivitate)

<https://github.com/PelleRemus/Facultate/blob/master/Anul%20I/Sem%20II/Algoritmi%20Fundamentali/MatriceScurtata/MatriceScurtata/Program.cs>.

1. Multimi de puncte: triunghiul de arie minima si acoperirea convexa (orice algoritm)

<https://github.com/PelleRemus/Facultate/tree/master/Anul%20I/Sem%20II/Algoritmi%20Fundamentali/%23Laborator/Triunghi%20arie%20minima/Triunghi%20arie%20minima>.

<https://github.com/PelleRemus/Facultate/tree/master/Anul%20I/Sem%20II/Geometrie/Seminar%205%20-%20Proiect/Graham%20Algorythm/Graham%20Algorythm> (rugati-va sa nu va ceara asa ceva).

1. Aria unui poligon oarecare

<https://github.com/PelleRemus/Facultate/tree/master/Anul%20III/Sem%20II%202019-2020/Tehnici%20avansate%20de%20programare/AriePoligon>.

1. Cata apa poate tine un vector (orice metoda)

<https://github.com/PelleRemus/Facultate/tree/master/Anul%20I/Sem%20II/Algoritmi%20Fundamentali/Picaturi%20de%20apa/Picaturi%20de%20apa>.

1. Fibonacci (recursiv si dinamic)

<https://github.com/PelleRemus/Facultate/tree/master/%23Anul%20II/Sem%20I/Met.%20Avansate%20de%20Prog/Fibonacci_Dynamic/Fibonacci_Dynamic>.

1. Regiunile N,S,E,V ale unei matrice

<https://github.com/PelleRemus/Facultate/tree/master/Anul%20III/Sem%20II%202019-2020/Tehnici%20avansate%20de%20programare/NSEV/NSEV>.

1. Rotirea unei matrice pe diagonala principala sau pe cea secundara

<https://github.com/PelleRemus/Facultate/tree/master/Anul%20III/Sem%20II%202019-2020/Tehnici%20avansate%20de%20programare/RotireMatriceDiagonala/RotireMatriceDiagonala>..

1. Problema Spectacolelor (greedy)

<https://github.com/PelleRemus/Facultate/tree/master/Anul%20I/Sem%20II/Algoritmi%20Fundamentali/Problema%20spectacolelor/Problema%20spectacolelor>

1. Bancnote (greedy)

<https://github.com/PelleRemus/Facultate/tree/master/Anul%20III/Sem%20II%202019-2020/Tehnici%20avansate%20de%20programare/Bancnote/Bancnote>.

1. Numere romane (greedy)

<https://github.com/PelleRemus/Facultate/blob/master/Anul%20I/Sem%20II/Algoritmi%20Fundamentali/%23Laborator/Numere%20romane/Numere%20romane/Program.cs>.

1. Ciurul lui Eratostene

<https://github.com/PelleRemus/Facultate/tree/master/Anul%20I/Sem%20II/Algoritmi%20Fundamentali/Ciurul%20lui%20Eratostene/Ciurul%20lui%20Eratostene>.