

# Probleme folosind tehnicile de programare Greedy, Programare Dinamică, Backtracking, Divide et Impera

- **Greedy:**

1. [Egyptian Fraction](#)
2. [Policemen catch thieves](#)
3. [Fitting Shelves Problem](#)
4. [Minimum Swaps for Bracket Balancing](#)
5. [Assign Mice to Holes](#)
6. [Minimum product subset of an array](#)
7. [Maximum product subset of an array](#)
8. [Maximize array sum after K negations](#)
9. [Maximize Array sum after incrementing at most K elements followed by dividing array by X](#)
10. [Maximise product of each array element with their indices by rearrangement](#)
11. [Aprindere](#)
12. [Reactivi](#)
13. [Int](#)
14. [Bleach](#)
15. [Livada](#)

- **Backtracking:**

1. [Sudoku](#)
2. [N-Queens Problem](#)
3. [Tug of War](#)
4. [Combinatorial Sum](#)
5. [Combinari](#)
6. [Submultimi](#)
7. [Flip](#)

- **Divide et Impera:**

1. [Merge Sort](#)
2. [Quick Sort](#)
3. [Closest Pair of Points](#) sau [Cele mai apropiate puncte din plan](#)
4. [Count inversions](#)
5. [Maximum subarray sum](#)
6. [Majority Element in a sorted array](#)
7. [Numbers whose factorials end with n zeros](#)
8. [Count number of occurrences \(or frequency\) in a sorted array](#)

- **Programare dinamică:**

1. [Custi](#)
2. [Subset Sum Problem](#)

3. [Painting Fence Problem](#)
4. [Unbounded Knapsack Problem](#)
5. [Longest Palindromic Subsequence](#)
6. [Joctv](#)
7. [Redu](#)