

Tehnici Avansate de Programare

Universitatea Aurel Vlaicu Arad

Examen parțial

Rezolvarea se trimită pe email: beniamin.bogosel@uav.ro

Instrucțiuni:

(a) Răspunsurile la întrebările teoretice sunt obligatorii.

(b) Formatul documentelor trimise trebuie să fie urmatorul:

- Notebook Jupyter ce conține răspunsul la întrebările teoretice și coduri Python la întrebările de tip cod. Puteți porni de la un fișier cu extensia .ipynb folosit la laborator (de pe platformă). Cei care aleg alt format: penalizare 1 punct.

(d) **Punctaj:** 1 punct din oficiu. 4 puncte – teorie. 5 puncte – problemele.

(e) Deadline: 02/12/2025

(f) Plagiatul se pedepsește cu nota minimă. Fiți originali!

Întrebări de curs:

1. Dați exemple de trei aplicații ale algoritmilor de sortare. Discutați rapid complexitatea problemelor enumerate dacă se folosește sortarea sau nu.

2. În urma analizei făcute în curs și în Laboratorul 4, care este ordinea de performanță pentru algoritmi Insertion Sort, Quick Sort, Merge Sort și Heap Sort în funcție de n ? Discutați situațiile "worst case" și "average case".

3. Care este diferența majoră între metodele "divide and conquer" și programarea dinamică?

4. Dați câte un exemplu de problemă pentru care structurile de date enumerate mai jos ajută la rezolvarea eficientă sau la ameliorarea complexității:

- arbore binar
- listă tip stivă
- dicționar

Testare: Fiecare problemă trebuie să aibă cel puțin 3 teste de nivel de dificultate crescător (cel puțin două netriviale). Lipsa testării va fi penalizată.

Problema 1. Fiind dat un sir x_1, \dots, x_n de numere prețuri ale unui obiect în n zile consecutive, determinați indicii $1 \leq i < j \leq n$ pentru care diferența $|x_j - x_i|$ este maximală. Implementați cel puțin una dintre metodele prezentate în curs, alegând o metodă mai eficientă decât soluția brute force.

Problema 2. Să presupunem că avem un sir de n valori x_1, x_2, \dots, x_n și căutăm să răspundem rapid unor cereri de forma: pentru i și j dați, care este cea mai mică valoare în intervalul x_i, \dots, x_j .

(a) Imagineați o aplicație practică în care o astfel de cerere ar putea fi utilă.

(b) Propuneți o structură de date ce folosește $O(n^2)$ spațiu în memorie și răspunde la întrebare în timp $O(1)$.

(c) Propuneți o structură de date ce folosește $O(n)$ spațiu în memorie și răspunde la întrebare în timp $O(\log n)$.

Timpul folosit pentru construirea structurii de date este neglijat. Se presupune că această operație este efectuată o singură dată. În ambele cazuri descrise mai sus implementați algoritmul corespunzător.