

Sortări de bază și căutări

1 Sortări clasice

Exercițiul 1.1. (2p) Să se scrie un program pentru ordonarea crescătoare a unui șir de n numere folosind metoda Bubble – Sort (sortarea prin interschimbare).

Exercițiul 1.2. (2p) Să se scrie un program pentru ordonarea crescătoare a unui șir de n numere folosind sortarea prin inserție (Insertion Sort).

Exercițiul 1.3. (2p) Să se scrie un program pentru ordonarea descrescătoare a unui șir de n numere folosind sortarea prin selecție (Selection Sort).

2 Căutări clasice

Exercițiul 2.1. (0.5p)

Scrieți algoritmul pentru căutarea secvențială a unui element într-un șir de n numere.

Exercițiul 2.2. (2p)

Scrieți algoritmul pentru căutarea binară a unui element într-un șir ordonat, de n numere.

3 Probleme

Exercițiul 3.1. (1.5p)

Scrieți o metodă de sortare a unui șir de string-uri astfel încât toate anagramele să fie una lângă cealaltă.

Exercițiul 3.2. (1.5p)

Un circ include în programul său un număr în care acrobații stau unul deasupra celuilalt pe umerii anteriorului. Din motive practice, fiecare persoană trebuie să fie mai scundă și mai ușoară decât persoana de dedesubt. Fiind date greutatea și înălțimile persoanelor din circ, să se scrie un algoritm pentru a calcula cea mai lungă secvență posibilă dintr-un astfel de turn uman.

Exemplu:

Input: (ht, wt): (150, 55) (170, 55) (120, 50) (190, 90) (150, 50) (160, 55) (172, 95)

Output: Cea mai lunga secvență este de lungime 6 (din vârf spre bază):

(190, 90) (170, 55) (160, 55) (150, 55) (150, 50) (120, 50) sau

(172, 95) (170, 55) (160, 55) (150, 55) (150, 50) (120, 50)

Exercițiul 3.3. (1.5p)

Se dă un șir ordonat crescător cu n elemente care a fost rotit de un număr necunoscut de ori. Să se scrie un algoritm de complexitate $\mathcal{O}(\log n)$ pentru căutarea unui număr în vector.

Exemplu:

Input: Căutarea lui 5 în (15 16 19 20 25 1 3 4 5 7 10 14)

Output: 8

Exercițiul 3.4. (1.5p)

Se dă un șir ordonat de string-uri *sparse* (există numeroase șiruri vide). Să se scrie un algoritm eficient pentru căutarea unui șir dat.

Exemplu:

1) Căutarea cuvântului "da" în ("arena", "", "", "", "da", "", "", "la", "", "", "minge", "", "") va întoarce 4

2) Căutarea cuvântului "poarta" în ("arena", "", "", "", "da", "", "", "la", "", "", "minge", "", "") va întoarce -1

4 Observații

1. Punctajul maxim necesar obținerii notei 10 la laboratorul de structuri de date este de 60p (6 laboratoare * 10p, ultimul laborator fiind recapitulativ).
2. Punctajul necesar pentru Laboratorul 1 se poate obține prin realizarea exercițiilor clasice + o problemă din secțiunea 3.
3. Transmiterea laboratorului se realizează pe adresa de e-mail ruxandra.balucea@unibuc.ro. Denumirea exercițiilor va fi de tipul x_y_grupa_Nume_Prenume (exemplu: 2_2_141_Po-pescu_Ion). De asemenea, în vederea obținerii punctajului aferent laboratorului se va realiza o prezentare a 1-2 probleme la cerința laborantului (dintre cele transmise prin e-mail).

4. Prezentarea problemei (problemelor) se poate realiza în timpul laboratoarelor L și $L+1$ unde L este considerat laboratorul curent (în cazul de față laboratoarele 1 și 2).
5. Prezentarea se poate desfășura atât fizic în timpul laboratorului, cât și online printr-o programare stabilită anterior cu laborantul (pentru cazuri speciale: simptome COVID, probleme personale etc.)
6. Înglobarea rezolvării de la exercițiile din secțiunile 1 și 2 în problemele din secțiunea 3 atrage obținerea punctajului și pentru acestea (de exemplu, folosirea unei sortări de tip Bubble-Sort în problema 3.1 duce la obținerea punctajului de 3.5p).
7. Rezolvarea tuturor problemelor suplimentare atrage după sine un punctaj bonus. Acesta nu se va menține pentru nota finală obținută la această materie, dar vă poate scuti de rezolvarea unui laborator viitor.