



Fișă de lucru – Algoritm

Modul: Algoritm

Durață: 90 minute

Nivel: Începători Grup țintă:

Tineri surzi care învață despre date și programare

Partea 1: Încălzirea (Înțelegerea elementelor de bază)

1. Suma a două cifre

Problema sumei a două cifre

Calculați suma a două numere cu o singură cifră.

Input: Două numere cu o singură cifră.

Rezultat: Suma acestor numere.

Exemplu: $2 + 3 = 5$

Implementarea unui algoritm

- pseudocod - C++ - Java-Python

2. Produs maxim pereche cu perechi

Problema produsului maxim pereche pe perechi

Găsește produsul maxim al a două numere distincte într-o succesiune de numere întregi nenegative.

Input: O succesiune de numere întregi nenegative.

Producție: Valoarea maximă care poate fi obținută prin înmulțirea a două elemente diferite din secvență.

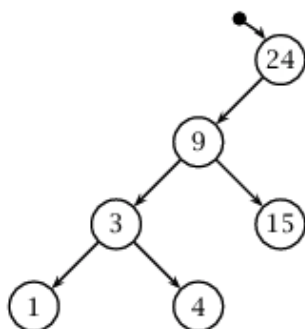
Exemplu:

	5	6	2	7	4
5		30	10	35	20
6	30		12	42	24
2	10	12		7	4
7	35	42	14		28
4	20	24	8	28	



Partea 2 — Practică: Activitate „arbore binar de căutare”

Luați în considerare următorul arbore binar de căutare (BST).



Întrebarea 1: Listați toate ordinele posibile de inserție (adică permutările) ale cheilor care ar fi putut produce acest BST.

Întrebarea 2: Trageți același BST după introducerea cheilor: 6, 45, 32, 98, 55 și 69, în această ordine.

Întrebarea 3: Extrageți BST rezultat din ștergerea cheilor 9 și 45 din BST, rezultate din întrebarea 2.

Întrebarea 4: Scrie cel puțin trei ordine de inserție (permutări) ale cheilor rămase în BST după întrebarea 3, care ar produce un arbore echilibrat (adică un arbore cu înălțime minimă).

Partea 3 — Exercițiu: „Sortare prin inserție”

Scrie pseudocodul algoritmului de inserție la loc și ilustrează execuția acestuia pe tablou.

$$A = (7, 17, 89, 74, 21, 7, 43, 9, 26, 10)$$

Fă asta scriind conținutul tabloului la fiecare iterație principală (exterioară) a algoritmului.



Partea 4 — Exersează: „max-heap”

Luați în considerare următorul max-heap

$H = 37, 12, 30, 10, 3, 9, 20, 3, 7, 1, 1, 7, 5$

Scrie ieșirea exactă a următorului algoritm `Extract-All` rulat pe H

`Extract-All (H)`

1 **în timp ce** $H.dimensiune$ de
grămadă > 0 2

`Heap-Extract-Max (H)`

3 **pentru** $i = 1$ **la** $H.heap-size$

4 **Producție** $H[i]$ 5

Producție “.” Sfârșitul liniei

`Heap-Extract-Max (H)`

1 **dacă** $H.dimensiune$ de
grămadă > 0 2 $k = H[1]$ 3

$H[1] = H[H.dimensiune$ de

grămadă] 4 $H.dimensiune$ de
grămadă =

$H.heap-size$ `Max-Heapify(` –
1

5 $H)$

6 **Întoarcerea** k



Partea 5 — Quiz rapid

Bifează (✓) răspunsul corect la fiecare întrebare.

1. Ce este un algoritm?
 - A. Un brand de calculatoare
 - B. Instrucțiuni pas cu pas pentru a rezolva o problemă
 - C. Un tip de date
 - D. O parolă
 2. Care dintre acestea este un exemplu cotidian de algoritm?
 - A. O rețetă de gătit
 - B. O fotbal
 - C. Un videoclip muzical.
 - D. O fotografie
 3. Cum numim informațiile pe care le introducem într-un algoritm?
 - A. Ieșire
 - B. Input
 - C. Eroare
 - D. Diagramă de flux
 4. Ce se întâmplă dacă un algoritm are o greșeală (un bug)?
 - A. Funcționează mai repede.
 - B. Se oprește sau dă rezultate greșite
 - C. Devine mai sigur.
 - D. Dispare
 5. Care este ordinea corectă a unui algoritm?
 - A. Ieșirea → proces → intrare
 - B. Proces → Intrare → Ieșire
 - C. Proces → Intrare → Ieșire
 - D. Intrare → Ieșire → Proces
-



Partea 6 — Reflecție

Răspunde cu fraze scurte sau propoziții.

1. Ce ai învățat astăzi despre **algoritmi**?

2. Ce a fost **dificil** pentru tine?
