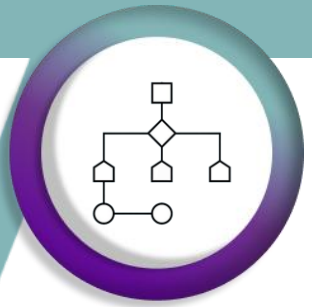
ALGORITM

DEAF YOUNG CODE

Număr proiect: 2023-2-IT03-KA220-YOU-000179130

Co-funded by
the European Union

Plan avansat de lecție



- Definește un algoritm într-un sens tehnic și explică-l simplu.
  Înțelege eficiența algoritmilor: află ce înseamnă "complexitatea timpului" și "complexitatea spațiului".
  Identificați tipurile comune de algoritmi: sortare, căutare, recursiv, iterativ, lacom etc.
  Analizează un algoritm pas cu pas pentru a verifica corectitudinea și eficiența.
  Scrie algoritmi simpli în pseudocod sau diagrame de flux.

Co-funded by
the European Union

Video

DEAF YOUNG CODE

Co-funded by the European Union

Ce este un algoritm?

DEAF YOUNG CODE

Definiție tehnică:

Un algoritm este o secvență finită de instrucțiuni bine definite pentru a rezolva o problemă.

Co-funded by the European Union

Caracteristici cheie

DEAF YOUNG CODE

1. Intrare: Datele cu care pornești.
2. Ieșire: Rezultatul sau soluția.
3. Finalitate: Trebuie să termini după un număr limitat de pași.
4. Certitudine: Fiecare pas este clar definit.
5. Eficiență: Fiecare pas poate fi efectuat într-un timp rezonabil.

Bazele algoritmilor

Cum funcționează algoritmii

- **Definește clar problema**→ Ce vrem să rezolvăm?
- **Proiectează o procedură pas cu pas**→ Care sunt instrucțiunile pentru a ajunge la o soluție?
- **Testează cu exemple**→ Verifică dacă algoritmul funcționează cu date de probă.
- **Analizează eficiența**→ Cât de rapid este? Câtă memorie consumă?
- **Redenumește și optimizează**→ Poate fi mai rapid sau să folosească mai puțină memorie?

Diagramă algoritmică(Flowchart)

O diagramă de flux este o modalitate vizuală de a prezenta un algoritm pas cu pas.
Te ajută să vezi logica, nu doar să o citești.

Element	Simbol	Function	Exemplu
Început / Sfârșit	Oval 	Arată unde începe sau se termină algoritmul	“Start” / “End”
Process / Step	Dreptunghi 	O acțiune sau instrucțiune	“Add two numbers”
Decizie / Condiție	Diamond 	O întrebare care schimbă fluxul (Da/Nu)	“Is a > b?”
Intrare / Ieșire	Parallelogram 	Date care intră sau ies	“Input numbers” / “Output result”
Arrow	→ 	Arată direcția curgerii	
Loop / Conector	Săgeată curbată sau etichetă 	Un pas care se repetă până când o condiție este îndeplinită	“Repeat until sorted”

Diagrama algoritmului (diagrama)

DEAF YOUNG CODE

Cum să citești o diagramă

- Start → Găsește ovalul “Start” .
- Input → Găsește ce infromații intră în sistem (de exemplu: numere, text).
- Procesul → Urmărește fiecare dreptunghi. Aceștia sunt pașii de urmat.
- Decizia → Când ai ajus la un diamant vei primi o întrebare Da/Nu

  Dacă “Yes,” urmărește săgeata.

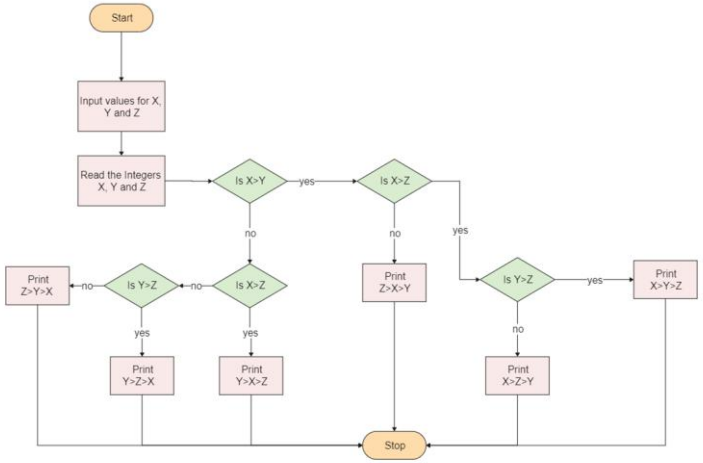
  If “No,” urmează următorii pași.
- Loop → Când o săgeată de duce la un pas în urmă înseamnă ”repetă”
- Output → La sfârșit, vezi rezultatul sau răspunsul.
- End → Prosesul se oprește.

Exemple: Sortarea Numerelor

Problem: Sort the numbers [4, 2, 7] from smallest to largest

1. Flowchart (Step-by-Step Visual)

```
Start
  |
  v
Input numbers: 4, 2, 7
  |
  v
Compare first two numbers:
Is 4 > 2? Yes → Swap
  |
  v
Numbers now: 2, 4, 7
  |
  v
Compare next numbers: 4 and 7
Is 4 > 7? No → Keep order
  |
  v
Output sorted numbers: 2, 4, 7
  |
  v
End
```



Pentru elevii surzi:

- **Folosește săgețile și casetele la fiecare pas.**
- **Evidențiază fiecare acțiune swap cu o culoare diferită.**
- **arată intrare → procesul → ieșire clară.**

Co-funded by the European Union

Exemple: Sortarea numerelor

Problem: Sort the numbers [4, 2, 7] from smallest to largest

2. Pseudocod (Versiune text simplă)

```
Algorithm SortThreeNumbers:
  Input: a, b, c
  If a > b then
      Swap a and b
  If b > c then
      Swap b and c
  If a > b then
      Swap a and b
  Output: a, b, c
End Algorithm
```

Notes:

- **Pseudocodul este ca instrucțiunile în engleză, nu ca limbajul de programare.**
- **Claritatea pas cu pas este esențială.**

Co-funded by the European Union

Exemple: Sortarea numerelor

Problem: Sort the numbers [4, 2, 7] from smallest to largest

3. Diagrama (Prezentare vizuală)

```
[Input: 4,2,7]
      |
      v
[Compare 4 and 2]
      |
      v
[Swap if needed] ---> [Numbers now: 2,4,7]
      |
      v
[Compare 4 and 7]
      |
      v
[Swap if needed] ---> [Numbers now: 2,4,7]
      |
      v
[Output: 2,4,7]
```

Co-funded by
the European Union

Puncte cheie

- Diagramă = hartă detaliată, pas cu pas, a algoritmului.
  - Include decizii, bucle, intrări, ieșiri.
  - Excelent pentru scrierea de pseudocod sau programare mai târziu.
- Diagramă = vizual de ansamblu, prezentare simplă.
  - Arată acțiunile principale și fluxul fără toți pașii.
  - Ajută la înțelegerea rapidă a logicii, mai ales pentru cei care învață vizual.

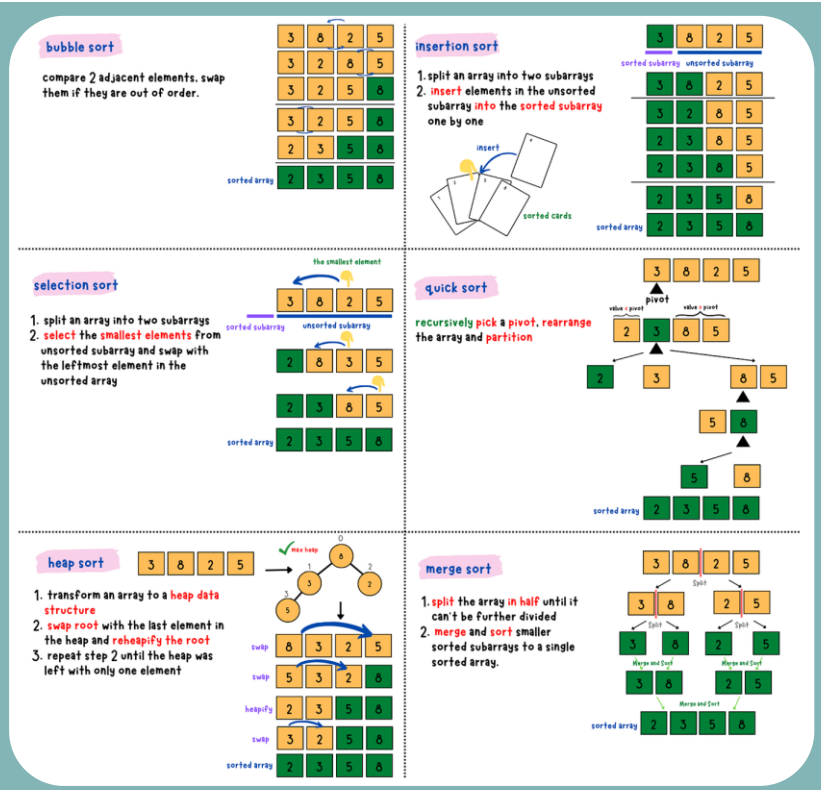
Co-funded by
the European Union

Teoria importanței algoritmilor

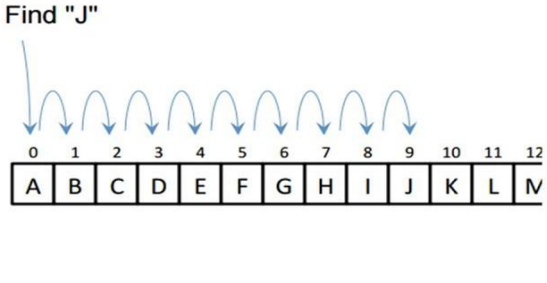
- **Complexitatea temporală (Big O notation): Măsoară cât de repede rulează un algoritm pe măsură ce dimensiunea intrărilor crește.**
- **Complexitatea spațială: Măsoară câtă memorie folosește un algoritm.**
- **Tipuri comune de algoritmi:**
  - Sortare: Sortare prin bule, sortare prin fuziune, sortare rapidă.
  - Căutare: Căutare liniară, căutare binară.
  - Recursivitate: Algoritmul se autonumește (e.g., factorial).
  - Greedy Algorithms: Fă cea mai bună alegere la fiecare pas (e.g., cea mai scurtă cale).
  - Programare dinamică: Împarte problemele în subprobleme mai mici.
- **Structuri de date: Algoritmii au adesea nevoie de liste, săgeți, arbori sau grafuri.**

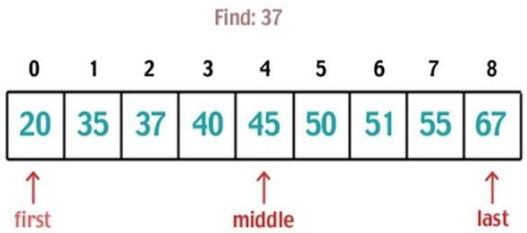
Co-funded by
the European Union

Sortare bazată pe comparație

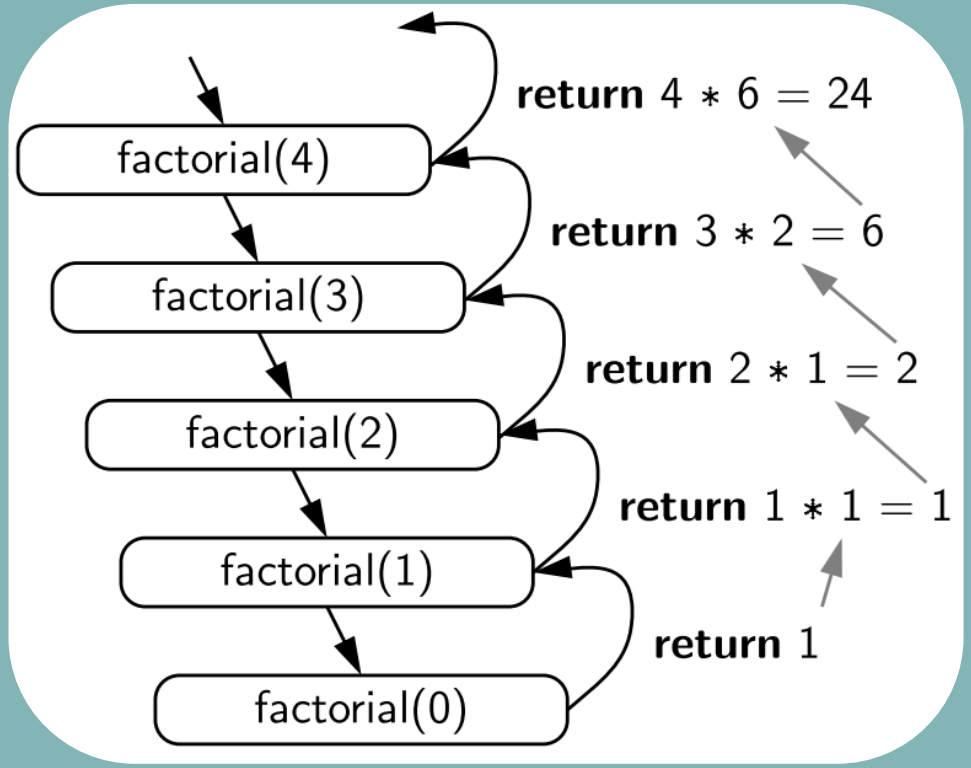


Sortare bazată pe comparație

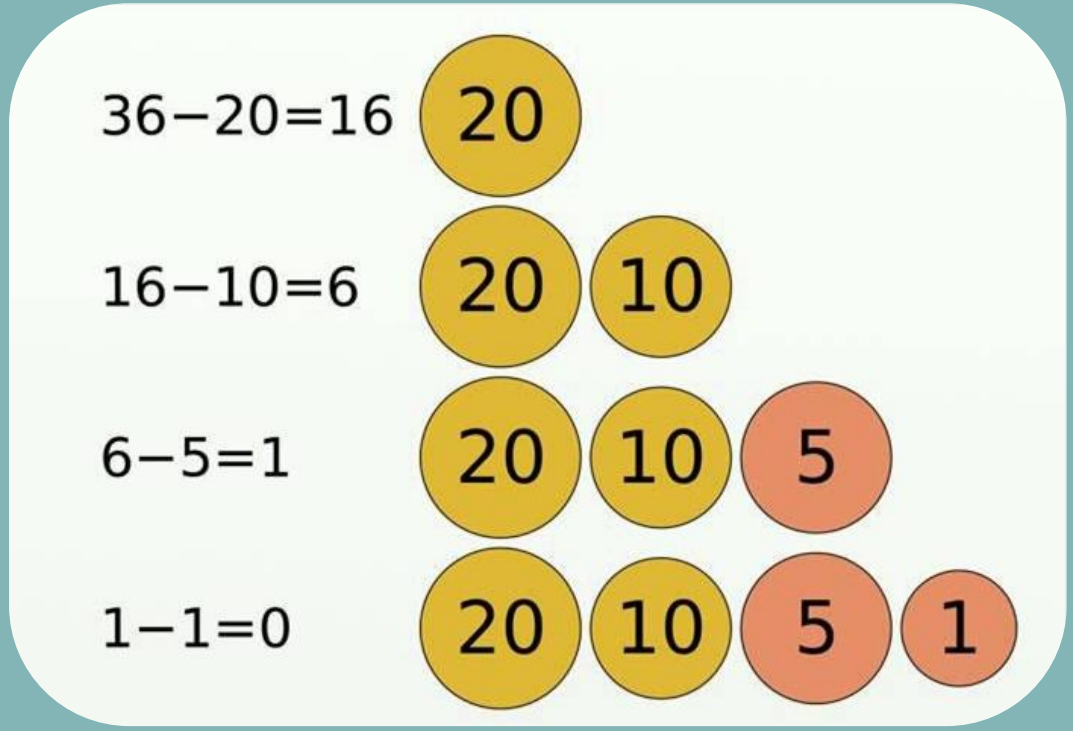




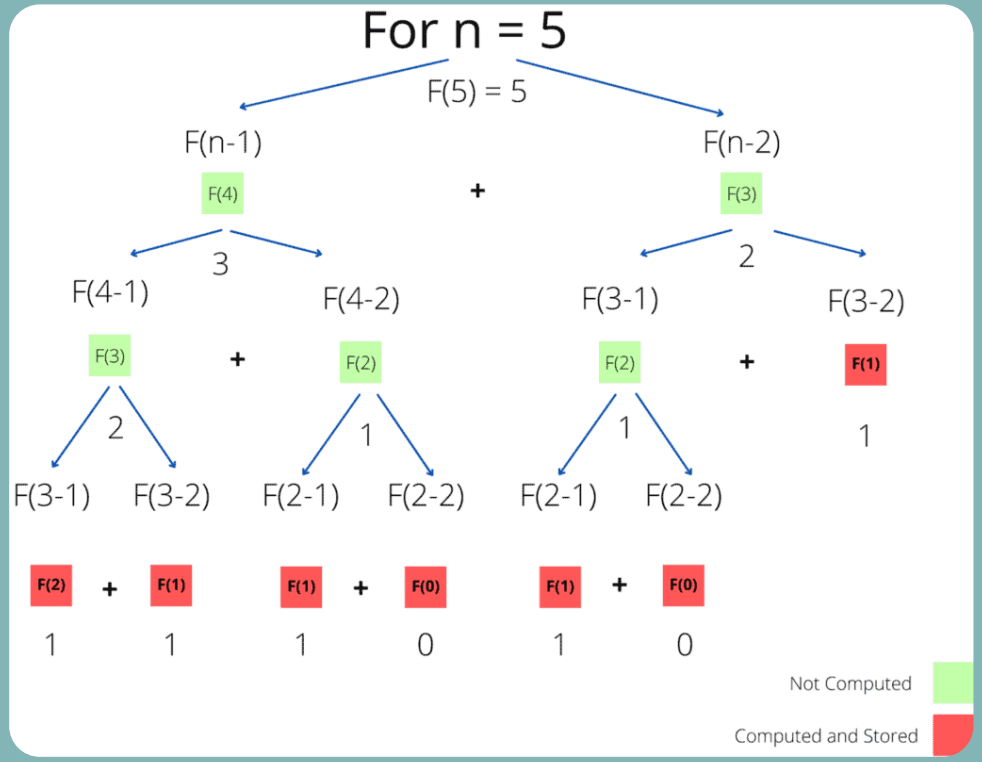
Sortare bazată pe comparație



Sortare bazată pe comparație



Sortare bazată pe comparație



Idee de exercițiu interactiv (învățare practică)

DEAF YOUNG CODE

Obiective

- Ajută studenții să exerseze proiectarea algoritmilor într-un mod distractiv și vizual.
- Întărește înțelegerea pașilor, deciziilor, buclelor și eficienței.

Idee de exercițiu interactiv (învățare practică)

DEAF YOUNG CODE

Pasul 1: Alege o sarcină simplă

Exemple de sarcini cotidiene:

- **Prepararea ceaiului**
  Sortarea cărților după înălțime
  Rutina de dimineață (să mă trezesc → să mă spăl pe dinți → să mănânc micul dejun → să mă îmbrac)
  Împachetarea unui rucsac
  Comandarea fructelor după mărime

Idee de exercițiu interactiv (învățare practică)

Pasul 2: Desenează o diagramă de flux

- Instrucțiuni pentru elevi:
- Start → Draw the start symbol (oval).
- Input / Output → Show the data or result (parallelogram).
- Actions / Steps → Use rectangles for each action.
- Decision Points → Use diamonds for Yes/No questions.
- Loops → Draw arrows that go back to repeat steps if needed.
- End → Show the final result or completion (oval).
- Tip: Use colors or sticky notes for actions, decisions, and outputs.

Idee de exercițiu interactiv (învățare practică)

Pasul 3: Distribuie și compară

- Roagă fiecare elev sau grup să-și prezinte diagrama de flux

Discuții:

- Au inclus toți pașii importanți?
  Există pași inutili care pot fi eliminați?
  Poate sarcina să fie realizată mai repede sau mai eficient?

Idee de exercițiu interactiv (învățare practică)

Pasul 4: Provocare Opțională (Eficiență)

- Introduce ideea eficienței algoritmului:
  - Mai puțini pași = algoritm mai rapid
  - Repetarea pașilor care nu sunt necesari = algoritm mai lent

Idee de exercițiu interactiv (învățare practică)

Pas 4: Provocare opțională (Eficiență)

- Exemplu: Sortare 3 Cărți
  - **Elev A:** Verifică toate perechile de două ori→ 6 Pași
  - **Elev B:** Doar schimburi inteligente→ 3 Pași
- Întreabă: Care algoritm este mai bun? De ce?
- Arată că chiar și sarcinile simple de zi cu zi pot avea algoritmi "mai eficienți".

Idee de exercițiu interactiv (învățare practică)

DEAF YOUNG CODE

Pas 5: Hands-On Bonus (Învățare kinestezică)

- Folosește obiecte reale: cărți, căni, post-it-uri.
- Elevii interpretează pașii fizic, urmându-și propriul flux.
- Ajută cursanții vizuali și tactili să înțeleagă conceptele: secvența, bucle și deciziile.

Vă mulțumesc pentru atenție!

DEAF YOUNG CODE

Rămâneți pe aproape pentru următoarea prezentare!

Număr proiect: 2023-2-IT03-KA220-YOU-000179130

LICEUL TEHNOLOGIC SPECIAL „VASILE PAVELCU“ - IAȘI

Co-funded by the European Union

CODE