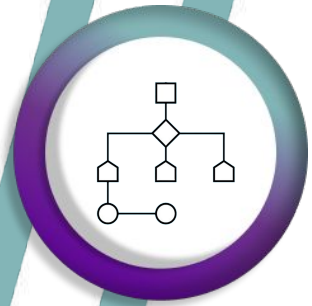
ALGORITMUS

DEAF YOUNG CODE

Číslo projektu: 2023-2-IT03-KA220-YOU-000179130

Co-funded by
the European Union

Pokročilý plán hodiny



- Definujte algoritmus v technickom zmysle a vysvetlite ho jednoducho.
- Porozumieť efektívnosti algoritmu: vedieť, čo znamenajú pojmy „časová zložitosť“ a „priestorová zložitosť“.
- Identifikujte bežné typy algoritmov: triedenie, vyhľadávanie, rekurzívne, iteratívne, chamtivé atď.
- Analyzujte algoritmus krok za krokom, aby ste skontrolovali jeho správnosť a efektívnosť.
- Napíšte jednoduché algoritmy v pseudokóde alebo vo forme vývojových diagramov

Co-funded by
the European Union

Video

DEAF YOUNG CODE

Co-funded by
the European Union

Čo je to algoritmus?

DEAF YOUNG CODE

Technická definícia:

Algoritmus je konečná sekvencia jasne definovaných inštrukcií na riešenie problému.

**Ps. Pamätajte, že riešenie nie je „je
“.**

Co-funded by
the European Union

Kľúčové charakteristiky

DEAF YOUNG CODE

1. Vstup: Údaje, s ktorými začínate.
2. Výstup: Výsledok alebo riešenie.
3. Konečnosť: Musí sa dokončiť po obmedzenom počte krokov.
4. Určitosť: Každý krok je jasne definovaný.
5. Účinnosť: Každý krok je možné vykonať v primeranom čase. time.

Základy algoritmov

DEAF YOUNG CODE

Ako fungujú algoritmy?

- **Jasne definujte problém** → Čo chceme vyriešiť?
- **Navrhnite postup krok za krokom** → Aké sú pokyny na dosiahnutie riešenia?
- **Test s príkladmi** → Skontrolujte, či algoritmus funguje s vzorovými vstupmi.
- **Analyzovať efektívnosť** → Ako rýchlo to funguje? Koľko pamäte to používa?
- **Vylepšiť a optimalizovať** → Môže to byť rýchlejšie alebo používať menej pamäte?

Algoritmický diagram (Flowchart)

DEAF YOUNG CODE

Funkčný diagram je vizuálny spôsob, ako znázorniť algoritmus krok za krokom. Pomáha vám to pochopiť logiku, nielen ju čítať

Element / Prvok	Symbol	Function / Funkcia	Príklad
Štart / Koniec	Ovál 	Ukazuje, kde algoritmus začína alebo končí.	“Start” / “End”
Proces / Krok	Obdĺžnik 	Akcia alebo pokyn	“Add two numbers”
Rozhodnutie / Podmienka	Diamant 	Otázka, ktorá mení priebeh (Áno/Nie)	“Is a > b?”
Vstup / výstup	Paralelogram 	Dáta, ktoré prichádzajú alebo odchádzajú	“Input numbers” / “Output result”
Šípka	→ 	Ukazuje smer toku	
Slučka / Konektor	Zakrivené šípky alebo štítky 	Krok, ktorý sa opakuje, kým nie je splnená podmienka	“Repeat until sorted”

Algoritmický diagram (flowchart)

DEAF YOUNG CODE

Ako čítať vývojový diagram

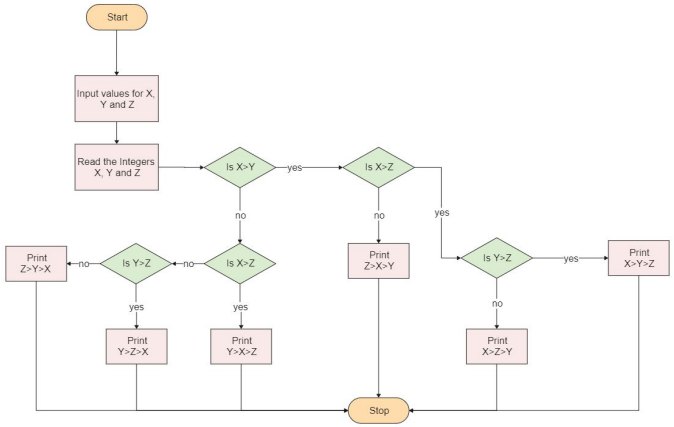
- Štart → Nájdite ovál „Štart“.
- Vstup → Pozrite sa, aké informácie vstupujú do systému (napríklad: čísla, text).
- Proces → Postupujte podľa jednotlivých obdĺžnikov. Ide o kroky alebo úkony.
- Rozhodnutie → Keď dosiahnete diamant, zobrazí sa otázka s možnosťou odpovede Áno/Nie.
  - Ak „Áno“, postupujte podľa jednej šípky.
  - Ak „Nie“, postupujte podľa ďalšieho pokynu.
- Slučka → Keď šípka smeruje späť k predchádzajúcemu kroku, znamená to „opakovať“.
- Výstup → Na konci si pozrite výsledok alebo odpoveď.
- Koniec → Proces sa zastaví

Príklady: Triedenie čísel

Problém: Zoradiť čísla [4, 2, 7] od najmenšieho po najväčšie.

1. Flowchart (Krok za krokom vizuálne)

```
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
```



Tip pre nepočujúcich študentov:

- **Pre každý krok použite šípky a rámčeky.**
- **Zvýrazniť akciu výmeny inou farbou.**
- **Zobraziť vstup → proces → výstup jasne.**

Co-funded by the European Union

Príklady: Triedenie čísel

Problém: Zoradiť čísla [4, 2, 7] od najmenšieho po najväčšie.

2. Pseudokód (jednoduchá textová verzia)

```
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
```

Poznámky:

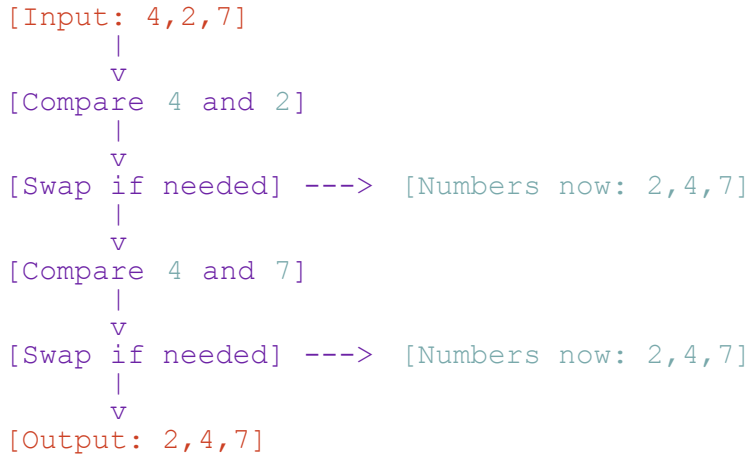
- **Pseudokód je ako pokyny v angličtine, nie programovací jazyk.**
- **Kľúčom je jasnosť v každom kroku.**

Co-funded by the European Union

Príklady: Triedenie čísel

Problém: Zoradiť čísla [4, 2, 7] od najmenšieho po najväčšie.

3. Diagram (vizuálny prehľad))



Co-funded by
the European Union

Kľúčové body

- Flowchart = podrobný, krok za krokom mapujúci algoritmus.
  - Zahŕňa rozhodnutia, slučky, vstupy, výstupy.
  - Vynikajúce na písanie pseudokódu alebo neskoršie kódovanie.

- Diagram = veľký vizuálny prehľad, jednoduchý
  - Zobrazuje hlavné akcie a priebeh bez všetkých krokov.
  - Pomáha rýchlo pochopiť logiku, najmä pre vizuálnych študentov.

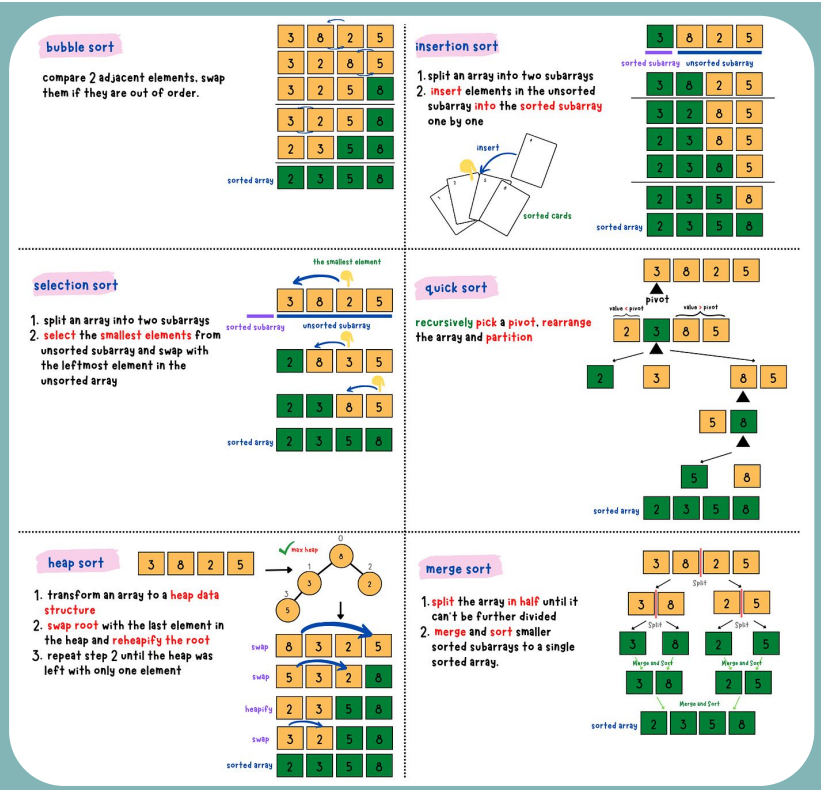
Co-funded by
the European Union

Teória algoritmov - dôležité

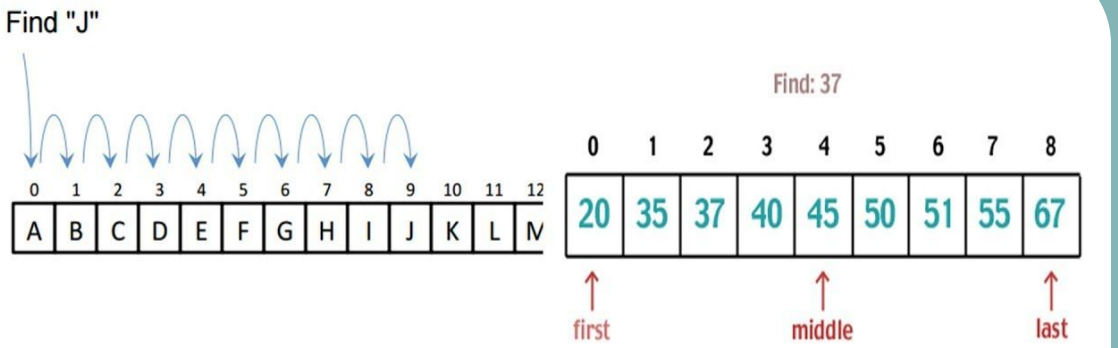
- **Časová zložitosť (označenie Big O):** Meranie rýchlosti algoritmu pri rastúcej veľkosti vstupných údajov**.**
- **Priestorová zložitosť:** Meranie množstva pamäte, ktorú algoritmus používa.
- **Bežné typy algoritmov:**
  - Triedenie: Bubble sort, Merge sort, Quick sort.
  - Vyhľadávanie: Lineárne vyhľadávanie, binárne vyhľadávanie.
  - Rekurzia: Algoritmus volá sám seba (napr. faktoriál).
  - Chamtivé algoritmy: V každom kroku urobte najlepšiu voľbu (napr. najkratšia cesta).
  - Dynamické programovanie: Rozdeľte problémy na menšie čiastkové problémy.
- **Dátové štruktúry:**
  Algoritmy často potrebujú zoznamy, polia, stromy alebo grafy.

Co-funded by
the European Union

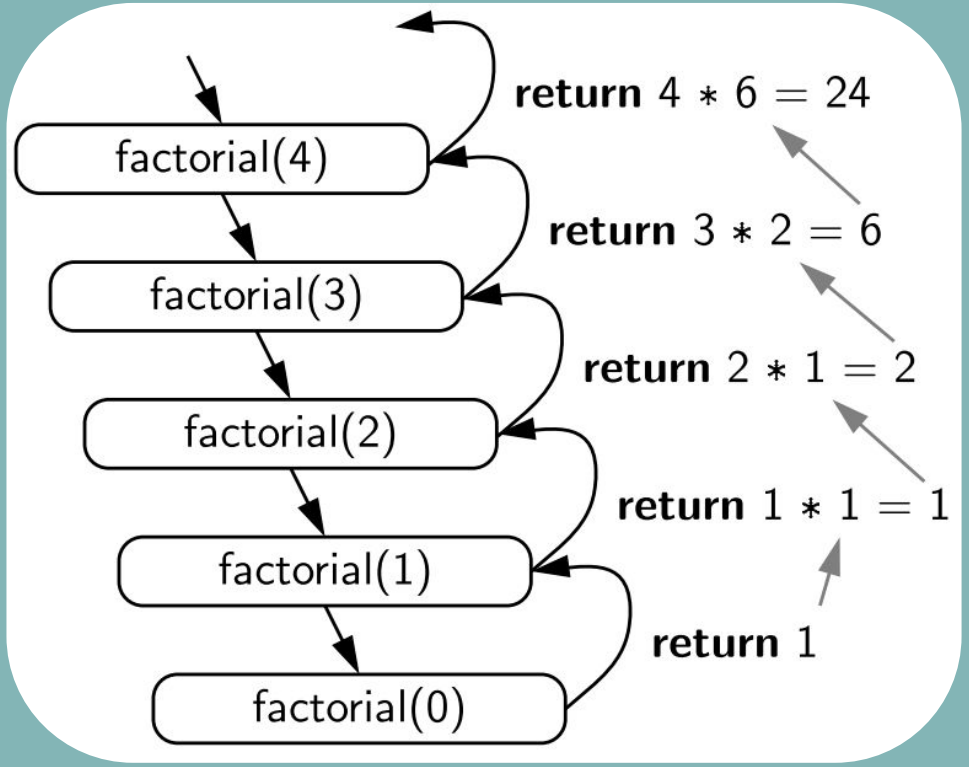
Triedenie na základe porovnania



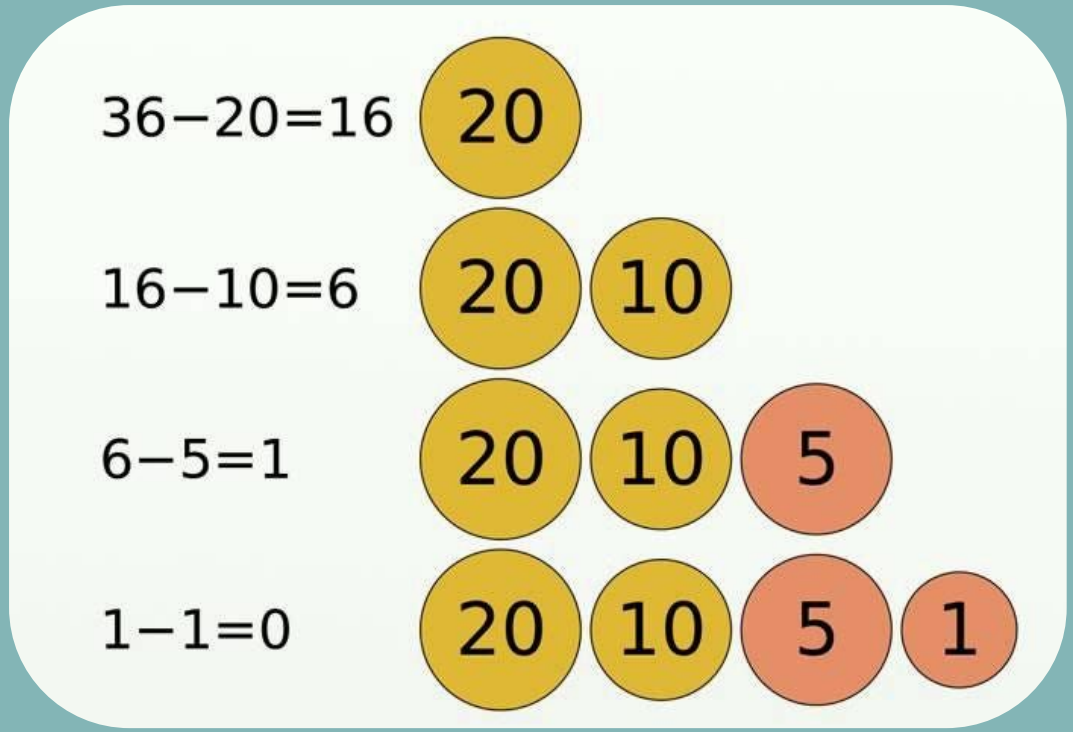
Triedenie na základe porovnania



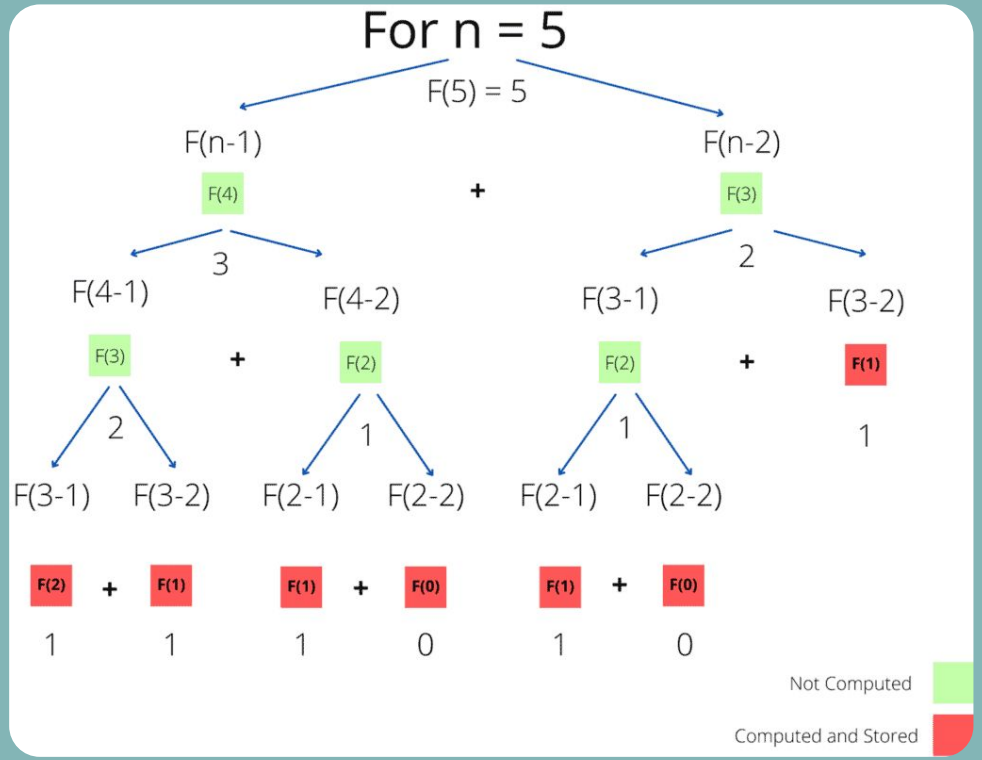
Triedenie na základe porovnania



Triedenie na základe porovnania



Triedenie na základe porovnania



Nápad na interaktívne cvičenie (praktické učenie)

DEAF YOUNG CODE

Cieľ

- Pomôžte študentom precvičovať návrh algoritmov zábavným a vizuálnym spôsobom.
- Posilniť pochopenie krokov, rozhodnutí, slučiek a efektívnosti

Nápad na interaktívne cvičenie (praktické učenie)

DEAF YOUNG CODE

Krok 1: Vyberte si jednoduchú úlohu

Príklady každodenných úloh:

- **Príprava čaju**
- **Ranná rutina (vstávanie → umývanie zubov → raňajky → obliekanie)**
- **Balenie batohu**
- **Triedenie ovocia podľa veľkosti**

Nápad na interaktívne cvičenie (praktické učenie)

Krok 2: Nakreslite diagram toku

- Pokyny pre študentov:
  Začiatok → Nakreslite symbol začiatku (ovál).
- Vstup / Výstup → Zobraziť údaje alebo výsledok (paralelogram).
- Akcie / kroky → Pre každú akciu použite obdĺžniky.
- Rozhodovacie body → Pre otázky typu áno/nie použite diamanty.
- Slučky → Nakreslite šípky, ktoré sa vracajú späť, aby sa kroky v prípade potreby opakovali.
- Koniec → Zobraziť konečný výsledok alebo dokončenie (ovál).

Tip: Používajte farby alebo lepiace poznámky pre akcie, rozhodnutia a výstupy.

Nápad na interaktívne cvičenie (praktické učenie)

Krok 3: Zdieľajte a porovnávajte

- Požiadajte každého študenta alebo skupinu, aby predstavili svoj diagram.

Diskutujte:

- Zahrnuli všetky dôležité kroky?
- Existujú zbytočné kroky, ktoré je možné odstrániť?
- Je možné úlohu vykonať rýchlejšie alebo efektívnejšie?

Nápad na interaktívne cvičenie (praktické učenie)

Krok 4: Voliteľná úloha (efektívnosť)

Predstavte pojem efektívnosť algoritmu:

- Menej krokov = rýchlejší algoritmus
- Opakovanie zbytočných krokov = pomalší algoritmu

Nápad na interaktívne cvičenie (praktické učenie)

Krok 4: Voliteľná úloha (efektívnosť)

- Príklad: Triedenie 3 kníh
  - **Študent A:** Skontroluje všetky páry dvakrát → 6 krokov
  - **Študent B:** Iba inteligentné výmeny → 3 kroky
- Otázka: **Ktorý algoritmus je lepší? Prečo**
- Ukážte, že **aj jednoduché každodenné úlohy** môžu mať „efektívnejšie“ algoritmy

Nápad na interaktívne cvičenie (praktické učenie)

Krok 5: Praktický bonus (kinestetické učenie)

- Používajte **skutočné predmety**: knihy, šálky, lepiace lístočky.
- Študenti **predvádzajú kroky** fyzicky, podľa vlastného diagramu toku
- Pomáha vizuálnym a hmatovým študentom pochopiť sekvencie, slučky a rozhodnutia

Ďakujeme za pozornosť!

DEAF YOUNG CODE

Sledujte ďalšiu prezentáciu!

Financované Európskou úniou. Uvedené názory a stanoviská sú však výlučne názormi autora (autorov) a nemusí sa zhodovať s názormi Európskej únie. Európskej únie alebo Európskej výkonnej agentúry pre vzdelávanie a kultúru (EACEA). Európska únia ani EACEA za ne nenesú žiadnu zodpovednosť.

Číslo projektu: 2023-2-IT03-KA220-YOU-000179130

Co-funded by the European Union