

Pracovný list – Algoritmus

Modul: Algoritmus

Trvanie: 90 minút

Úroveň: Začiatočník

Cieľová skupina: Nepočujúci mladí ľudia, ktorí sa učia o dátach a programovaní

Časť 1: Rozcvička (Porozumenie základom)

1. Sčítanie dvoch čísiel

Úloha: Sčítanie dvoch čísiel

Vypočítajte súčet dvoch jednomiestnych čísiel.

Vstup: Dve jednomiestne čísla.

Výstup: Súčet týchto čísiel.

Príklad: $2 + 3 = 5$

Implementácia algoritmu

- Pseudokód
- C++
- Java
- Python

2. Problém maximálneho súčinu dvojíc

Úloha o maximálnom súčine dvoch čísiel

Nájdite maximálny súčin dvoch rôznych čísiel v postupnosti nezáporných celých čísiel.

Vstup: Postupnosť nezáporných celých čísiel.

Výstup: Maximálna hodnota, ktorú možno získať vynásobením dvoch rôznych prvkov z postupnosti.

Príklad:

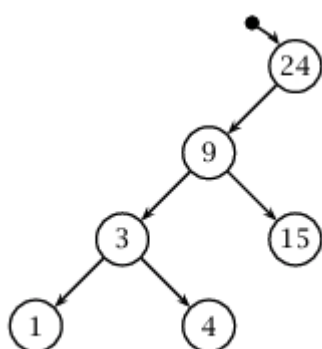
	5	6	2	7	4
5		30	10	35	20
6	30		12	42	24
2	10	12		7	4
7	35	42	14		28
4	20	24	8	28	

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

Financované Európskou úniou. Uvedené názory a stanoviská sú však výlučne názormi autora (autorov) a nemusia nevyhnutne odzrkadľovať názory Európskej únie ani Európskej výkonnej agentúry pre vzdelávanie a kultúru (EACEA). Za tieto názory a stanoviská nemôže niesť zodpovednosť ani Európska únia, ani EACEA.

Časť 2 — Cvičenie: Aktivita „Binárny vyhľadávací strom“

Zvažte nasledujúci binárny vyhľadávací strom (BST).



Otázka 1: Vymenujte všetky možné poradia vloženia (t. j. permutácie) kľúčov, ktoré mohli vytvoriť tento BST.

Otázka 2: Nakreslite ten istý BST po vložení kľúčov: 6, 45, 32, 98, 55 a 69, v tomto poradí.

Otázka 3: Nakreslite BST, ktorý vznikne odstránením kľúčov 9 a 45 z BST vzniknutého v otázke 2.

Úloha 4: Napíšte aspoň tri poradia vkladania (permutácie) kľúčov, ktoré zostali v BST po úlohe 3, ktoré by vytvorili vyvážený strom (t. j. strom s minimálnou výškou).

Časť 3 — Cvičenie: „Vkladacie triedenie“

Napište pseudokód algoritmu vkladacieho triedenia na mieste a znázorníte jeho vykonanie na poli

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

Urobte to tak, že v každej hlavnej (vonkajšej) iterácii algoritmu zapíšete obsah poľa.

Časť 4 — Cvičenie: „max-heap“

Zvažte nasledujúcu max-heap

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

Napište presný výstup nasledujúceho algoritmu `Extract-All` spusteného na `H`

`Extract-All (H)`

```
1 while H.heap-size > 0
2   Heap-Extract-Max (H)
3   pre i=1 až H.heap-size
4     výstup H[i]
5   vypíše „“ na koniec
riadku
```

`Heap-Extract-Max (H)`

```
1, ak H.heap-size > 0
2   k = H[1]
3   H[1] = H[H.heap-size]
4   H.velkost'-haldy =
H.velkost'-haldy-1
5   Max-Heapify (H)
6 vráti k
```

Časť 5 — Rýchly kvíz

Zaškrtnite (✓) správnu odpoveď pri každej otázke.

1. Čo je algoritmus?
 - A. Značka počítača
 - B. Postupné pokyny na riešenie problému
 - C. Typ údajov
 - D. Heslo

2. Ktorý z nasledujúcich príkladov je bežným príkladom algoritmu?
 - A. Kuchársky recept
 - B. Futbalová lopta
 - C. Hudobný videoklip
 - D. Obrázok

3. Ako nazývame informácie, ktoré vkladáme do algoritmu?
 - A. Výstup
 - B. Vstup
 - C. Chyba
 - D. Prehľadový diagram

4. Čo sa stane, ak algoritmus obsahuje chybu (bug)?
 - A. Funguje rýchlejšie
 - B. Zastaví sa alebo poskytne nesprávne výsledky
 - C. Stane sa bezpečnejším
 - D. Zmizne

5. Aké je správne poradie krokov v algoritme?

- A. Výstup → Spracovanie → Vstup
- B. Spracovanie → Vstup → Výstup
- C. Vstup → Spracovanie → Výstup
- D. Vstup → Výstup → Spracovanie

Časť 6 — Reflexia

Odpovedzte krátkymi frázami alebo vetami.

1. Čo ste sa dnes naučili o **algoritme**?

2. Čo bolo pre vás **ťažké**?
