

## Munkalap – Algoritmus

**Modul:** Algoritmus

**Időtartam:** 90 perc

**Szint:** Kezdő

**Célcsoport:** Siket fiatalok, akik az adatokról és a kódolásról tanulnak

---

### 1. rész: Bemelegítés (az alapok megértése)

#### 1. Két számjegy összege

**Két számjegyű szám összegének feladata**

*Számítsd ki két egyjegyű szám összegét.*

**Bemenet:** Két egyjegyű szám.

**Kimenet:** Ezen számok összege.

**Példa:**  $2 + 3 = 5$

**Algoritmus implementálása**

- Pszeudokód
- C++
- Java
- Python

#### 2. Maximális páronkénti szorzat

**Maximális páronkénti szorzat feladat**

*Keresse meg két különböző szám maximális szorzatát egy nemnegatív egész számok sorozatában.*

**Bemenet:** Nemnegatív egész számok sorozata.

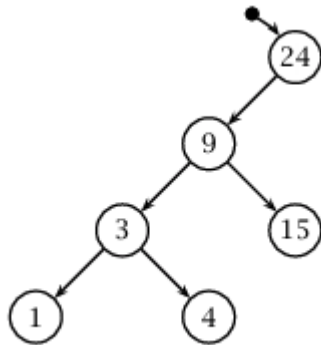
**Kimenet :** A sorozat két különböző elemének szorzatával előállítható maximális érték.

**Példa:**

|   | 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 |    |

## 2. rész – Gyakorlás: „ Bináris keresőfa ” feladat

Tekintsük a következő bináris keresőfát (BST).



1. kérdés : Sorolja fel a kulcsok összes lehetséges beszúrási sorrendjét (azaz permutációját), amely ezt a BST-t eredményezhette volna.

2. kérdés: Rajzold le ugyanazt a BST-t a kulcsok beillesztése után: 6, 45, 32, 98, 55 és 69, ebben a sorrendben.

3. kérdés: Rajzold le a 2. kérdésből származó BST-ből a 9-es és 45-ös kulcsok törlésével kapott BST-t!

4. kérdés: Írjon fel legalább három olyan beszúrási sorrendet (permutációt) a 3. kérdés után a BST-ben maradt kulcsokról, amelyek egy kiegyensúlyozott fát (azaz egy minimális magasságú fát) eredményeznének.

### 3. rész – Gyakorlás: „Beszúrásos rendezés”

Írd meg a helybeni beszúrásos rendezési algoritmus pszeudokódját, és szemléltesd a végrehajtását a tömbön!

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

Ezt úgy tedd meg, hogy a tömb tartalmát az algoritmus minden fő (külső) iterációjában kiírod.

---

### 4. rész – Gyakorlás: „max-heap”

Tekintsük a következő maximális halom függvényt:

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

Írd fel a következő, H-n futtatott Extract-All algoritmus pontos kimenetét:

Összes kinyerése ( H )

```
1 míg a H.heap-size >0
2   Heap-Extract-Max ( H )
3   i = 1 esetén H.heap-size
4       kimenet H[i]
5   kimenet “.” sorvég
```

```
Heap-Extract-Max ( H )
1 ha H.heap-size >0
2   k = H[1]
3   H[1] = H[H.heap-size ]
4   H.heap-size = H.
   heap-size -1
```

5      `Max-Heapify( H )`

6 **return** k

---

## 5. rész – Gyors kvíz

Jelöld be ( ✓ ) a helyes választ minden kérdésben.

1. Mi az az algoritmus?
  - A. Egy számítógép márka
  - B. Lépésről lépésre útmutató egy probléma megoldásához
  - C. Egy adattípus
  - D. Egy jelszó
2. Melyik az alábbiak közül egy mindennapi algoritmus példa?
  - A. Egy főzőrecept
  - B. Egy focilabda
  - C. Egy videoklip
  - D. Egy kép
3. Hogyan nevezzük az algoritmusba bevitt információkat?
  - A. Kimenet
  - B. Bemenet

- C. Hiba
  - D. Folyamatábra
4. Mi történik, ha egy algoritmusban hiba (bug) van?
- A. Gyorsabban működik
  - B. Leáll vagy rossz eredményeket ad
  - C. Biztonságosabbá válik
  - D. Eltűnik
5. Mi az algoritmus helyes sorrendje?
- A. Output → Process → Input
  - B. Process → Input → Output
  - C. Input → Process → Output
  - D. Input → Output → Process
- 

## 6. rész – Elmélkedés

Válaszolj rövid mondatokkal vagy kifejezésekkel.

1. Mit tanultál ma **az algoritmusokról** ?

---

2. Mi volt **nehéz** számodra?

---