

## 2. GYAKORLAT

### Áttekintő:

- Tudnivalók ismertetése (szabályok, osztályozás)
- Header és c fájl struktúra
- Algoritmusok bonyolultsága
- Egyszerű feladatok implementálása

### Kitűzött feladatok:

#### 1. Feladat:

Vizsgáld a **lineáris keresés** bonyolultságát. Írd ki a képernyőre, hogy mennyi a **végrehajtási idő** (mp-ben), illetve a **műveletek száma**. Teszteld  $n = 10, 100, 1000 \dots 1000000000$  elemű sorozatra is. A sorozat elemei és a keresett érték is legyen véletlenszerűen generálva. Alkalmazd a **.h** és **.c** fájl struktúrát.

#### 2. Feladat:

Vizsgáld a **bináris keresés** (rekurzív változat) bonyolultságát. Írd ki a képernyőre, hogy mennyi a **végrehajtási idő** (mp-ben), illetve a **műveletek száma**. Teszteld  $n = 10, 100, 1000 \dots 1000000000$  elemű sorozatra is. A sorozat elemei és a keresett érték is legyen véletlenszerűen generálva. Biztosítsd a sorozat elemeinek rendezettségét mielőtt alkalmaznád a bináris keresést. Használj **qsort**-ot vagy más általad választott **rendező algoritmust**. Alkalmazd a **.h** és **.c** fájl struktúrát.

#### 3. Feladat:

Implementáld a következő játékot: egy magyar kártya pakliból kihúzzunk egy lapot. Eldöntendő kérdések által kell kitaláljuk, hogy melyik a kihúzott lap.

Legkevesebb hány eldöntendő kérdésből tudjuk biztosan eldönteni, hogy melyik lapot húzta az illető?

#### 4. Feladat

Egy egyetemista a fesztiválon átbulizott éjszaka után elfelejti, hogy a kempingben hol is volt a sátra. A kempingben  $n$  sátorhely van, amik *egy sorban vannak* elhelyezve. Az

## 2. GYAKORLAT

egyetemista csak akkor ismeri fel a sátrát, mikor *másodszor* is alaposan megnézi, így a következő stratégiát alkalmazza: először elmegy a kemping másik széléig, miközben minden sátort megnéz. Majd, mivel nem találja a sátrát, visszafordul, és újra megvizsgálja az összes sátorhelyet, míg rá nem talál a sajátjára. Hányszor kell az egyetemistának megvizsgálni a sátrakat a legjobb, a legrosszabb és az átlagos esetben.

### 5. Feladat

Hasonlítsd össze a **lineáris**- és **bináris** keresés bonyolultságát (mindkettőt ugyanazokon a számsorozatokon teszteld  $n = 10, 100, 1000 \dots 1000000000$  esetén). Mit veszel észre?

A 3-as és 4-es feladathoz segítséget kapsz az alábbi linken keresztül: [alga-gyak-01.pdf \(u-szeged.hu\)](#)

### 6. Feladat

Határozd meg a következő algoritmus komplexitását:

```
for( int i = n; i > 0; i /= 2 )
{
    for( int j = 1; j < n; j *= 2 )
    {
        for( int k = 0; k < n; k += 2 )
        {
            ... // constant number of operations
        }
    }
}
```

### 7. Feladat:

Határozd meg a következő algoritmus komplexitását:

```
for( int i = n; i > 0; i-- )
{
    for( int j = 1; j < n; j *= 2 )
    {
        for( int k = 0; k < j; k++ )
```

## 2. GYAKORLAT

```
        {  
            ... // constant number of operations  
        }  
    }  
}
```

### 8. Feladat:

Találjunk és adjunk vissza egy „csúcsot” egy számsorozatból, ha létezik! Ellenőrizd a szélsőséges eseteket (pl. nincs eredmény, üres a tömb stb.), működjön a program bármilyen bemenetre. Törekedj többféle megoldásra, és ellenőrizd melyik hatékonyabb (pl. iteratív megoldás vs. rekurzív megoldás).

### Extra feladatok:

#### 1. Extra:

Az előző alpontokban megírt lineáris- és bináris keresés esetén teszteld az algoritmusok bonyolultságát és hasonlítsd össze őket úgy, hogy minden  $n$  – re keresd a sorozat legelső, középső, legutolsó, valamely belső elemét, illetve egy olyan értéket, ami nincs benne. Mit tapasztalsz? Vezesd táblázatba az algoritmusok bonyolultságára vonatkozó észrevételeid.

Segítség a táblázathoz:

| n          | Első elem |     |                 |     | Középső elem |     |                 |     | Utolsó elem |     |                 |     | Belső elem |     |                 |     | Hiányzó elem |     |                 |     |
|------------|-----------|-----|-----------------|-----|--------------|-----|-----------------|-----|-------------|-----|-----------------|-----|------------|-----|-----------------|-----|--------------|-----|-----------------|-----|
|            | Idő       |     | Műveletek száma |     | Idő          |     | Műveletek száma |     | Idő         |     | Műveletek száma |     | Idő        |     | Műveletek száma |     | Idő          |     | Műveletek száma |     |
|            | lin       | bin | lin             | bin | lin          | bin | lin             | bin | lin         | bin | lin             | bin | lin        | bin | lin             | bin | lin          | bin | lin             | bin |
| 10         |           |     |                 |     |              |     |                 |     |             |     |                 |     |            |     |                 |     |              |     |                 |     |
| 100        |           |     |                 |     |              |     |                 |     |             |     |                 |     |            |     |                 |     |              |     |                 |     |
| 1000       |           |     |                 |     |              |     |                 |     |             |     |                 |     |            |     |                 |     |              |     |                 |     |
| 10000      |           |     |                 |     |              |     |                 |     |             |     |                 |     |            |     |                 |     |              |     |                 |     |
| 100000     |           |     |                 |     |              |     |                 |     |             |     |                 |     |            |     |                 |     |              |     |                 |     |
| 1000000    |           |     |                 |     |              |     |                 |     |             |     |                 |     |            |     |                 |     |              |     |                 |     |
| 10000000   |           |     |                 |     |              |     |                 |     |             |     |                 |     |            |     |                 |     |              |     |                 |     |
| 100000000  |           |     |                 |     |              |     |                 |     |             |     |                 |     |            |     |                 |     |              |     |                 |     |
| 1000000000 |           |     |                 |     |              |     |                 |     |             |     |                 |     |            |     |                 |     |              |     |                 |     |

### Hasznos linkek:

- <https://www.jetbrains.com/help/clion/github.html>
- <https://randerson112358.medium.com/how-to-run-c-program-in-command-prompt-e435186cd162>
- <https://www.programiz.com/dsa/algorithm>
- <https://www.geeksforgeeks.org/linear-search/>
- <https://www.geeksforgeeks.org/binary-search/>