

Algoritmusok megadása, kiválasztásos rendezés

ALGORITMUSOK ÉS GRÁFOK

1. gyakorlat

2025.

- Adjunk algoritmust egy tömb legnagyobb elemének megkeresésére. Indokoljuk meg, hogy miért helyes az algoritmus, és adjunk felső becslést az algoritmus lépésszámára, ahol egy lépésnek az összehasonlításokat, az értékadásokat és az eredmény visszaadását tekintjük.
- Írjuk le pszeudokóddal az alábbi szöveges algoritmust egy tömb legkisebb elemének megkeresésére. Végigmegyünk a tömbben, és sorra összehasonlítjuk minden cella tartalmát az utána következővel. Ha az elől álló elem kisebb, mint a hátul álló, akkor megcseréljük őket. A végén kiírjuk az utolsó cella tartalmát.
 - Mutassuk meg, hogy az algoritmus helyes és adjunk felső becslést az algoritmus lépésszámára, ahol egy lépésnek az összehasonlításokat, a cseréket, és az eredmény visszaadását tekintjük.
 - Milyen felső becslést tudunk adni az algoritmus lépésszámára, ha egy csere három lépésnek számít?
- Az órán tanult kiválasztásos rendezést (lásd az alábbi kódot) futtatjuk a 8, 1, 10, 23, 7, 2, 9, 11, 4 tömbben.
 - Hogyan néz ki a tömb akkor, amikor a külső ciklus futása a $j = 3$ értékkel kezdődik el?
 - Hogyan néz ki a tömb akkor, amikor a külső ciklus futása a $j = 3$ értékkel véget ér?

Input: egy $n \geq 1$ darab különböző számot tartalmazó T tömb

Output: növekvően rendezett T tömb

```
ciklus:  $j$  fut 0-tól  $(n - 2)$ -ig
     $\min := T[j]$ 
     $\min\_hely := j$ 
    ciklus:  $i$  fut  $(j + 1)$ -től  $(n - 1)$ -ig
        ha  $T[i] < \min$ :
             $\min := T[i]$ 
             $\min\_hely := i$ 
    ciklus vége
     $csere(T[j], T[\min\_hely])$ 
ciklus vége
return  $T$ 
```

- Határozzuk meg az alábbi algoritmusok lépésszámát, ahol egy lépésnek egy * kiírását tekintjük.

(a) **Input:** egy $n \geq 2$ egész szám

```
ciklus:  $i$  fut 0-tól  $(n - 1)$ -ig
    ciklus:  $j$  fut 1-től  $n$ -ig
        print "*"
    ciklus vége
ciklus vége
```

(b) **Input:** egy $n \geq 2$ egész szám

```
ciklus:  $i$  fut 0-tól  $(n - 1)$ -ig
    ciklus:  $j$  fut  $(i + 1)$ -től  $n$ -ig
        print "*"
    ciklus vége
ciklus vége
```

5. Adjunk algoritmust egy legalább két elemű, csupa különböző számot tartalmazó tömb legkisebb és második legkisebb elemének megkeresésére. Indokoljuk meg, hogy miért helyes az algoritmus, és adjunk felső becslést az algoritmus lépésszámára, ahol egy lépésnek az összehasonlításokat, az értékadásokat és az eredmény visszaadását tekintjük.
6. Adjunk algoritmust, amely az összes eset végigvizsgálásával eldönti, hogy egy bemenetként kapott n -méretű T tömbben van-e
- (a) kettő (b) három
- olyan különböző cella, amikben szereplő számok összege éppen 100. Mutassuk meg, hogy az algoritmus helyes és adjunk felső becslést az algoritmus lépésszámára, ahol egy lépésnek az összehasonlításokat, az összeadásokat, az értékadásokat és az eredmény visszaadását tekintjük.
7. Az alábbi pszeudokód inputja két tömb: az n hosszú A tömb egész számokat tartalmaz, a szintén n hosszú B tömb pedig csupa 0-ból áll. Az eljárás outputja az M szám. Mit számol ki ez az algoritmus, azaz mi az M ?

Input: egy $n \geq 1$ darab egész számot tartalmazó A tömb, és egy n -méretű üres B tömb

$B[0] := A[0]$

ciklus: i fut 1-től $(n - 1)$ -ig

$B[i] := B[i - 1] + A[i]$

ciklus vége

$M := B[0]$

ciklus: j fut 1-től $(n - 1)$ -ig

ha $B[j] > M$:

$M := B[j]$

ciklus vége

return M

8. Van hat darab számítógépes programunk, és az alábbi táblázat mutatja, hogy ezek egy bemenetként kapott tetszőleges n -méretű tömbön hány lépést tesznek. A táblázat kitöltésével határozzuk meg, hogy a bemenet méretének megduplázásával hányszorosára nő az egyes algoritmusok lépésszáma.

Algoritmus sorszáma	Lépésszám tetszőleges n -méretű tömbön	Hány lépést tesz az algoritmus egy		Hányszorosára nőtt a lépésszám?
		k -méretű tömbön?	$(2k)$ -méretű tömbön?	
1.	n	k	$2k$	kétszeresére
2.	$5n$			
3.	n^2			
4.	$3n^2$			
5.	n^3			
6.	$4n^3$			

9. Van egy számítógépes programunk, ami egy bemenetként kapott tetszőleges, legalább 2 méretű tömbből két lépésben egy eggyel kisebb méretű tömböt készít, majd ezt rekurzívan folytatja (azaz a kapott kisebb tömbből újabb két lépéssel egy még eggyel kisebb tömböt készít, és így tovább). Az algoritmus az 1-méretű tömbökön egy lépést tesz. Határozzuk meg az algoritmus lépésszámát.
10. Van egy számítógépes programunk, ami egy bemenetként kapott tetszőleges n -méretű tömbön legfeljebb $L(n)$ lépést végez. Határozzuk meg az $L(n)$ függvényt, ha $L(1) = 1$ és tetszőleges $n \geq 2$ esetén $L(n) = 2 \cdot L(n - 1)$.