

A nagy ordó fogalma
ALGORITMUSOK ÉS GRÁFOK
2. gyakorlat
2025.

Definíció. Azt mondjuk, hogy egy algoritmus $L(n)$ lépésszáma $O(f(n))$, ha van olyan c pozitív konstans és n_0 pozitív küszöbérték, hogy minden $n \geq n_0$ esetén $L(n) \leq c \cdot f(n)$ teljesül.

1. Egy algoritmus lépésszáma $100 \cdot n^2 + 10^{10} \cdot n + 17$. Alkalmas c konstans és n_0 küszöbérték megadásával bizonyítsuk be, hogy ez egy $O(n^2)$ -es algoritmus.
2. Az alábbiakban négy algoritmus lépésszámát láthatjuk. Melyik esetekben igaz, hogy az algoritmus $O(n^2)$ -es? Amikor úgy véljük, hogy ez igaz, akkor megfelelő c konstans és n_0 küszöbérték megadásával bizonyítsuk is ezt be.

(a) $10n^2 - n \log_2 n$, (b) $n + n^2 + n^3$, (c) $100000 \log_2 n$, (d) $n^2 \log_2 n + 10^5$

3. Az alábbiakban három algoritmus lépésszámát láthatjuk. Ezek közül pontosan az egyikre igaz, hogy $O(n)$ -es. Válasszuk ki, hogy melyik ez a lépésszám, valamint megfelelő c konstans és n_0 küszöbérték megadásával bizonyítsuk is ezt be.

$2023^{2024} \cdot 2^n + 4n$, $17\sqrt{n} + \frac{1}{n} + 210$, $9n \cdot \frac{\log_2 n}{15} - 256$

4. Az alábbiakban három algoritmus lépésszámát láthatjuk. Ezek közül pontosan az egyikre igaz, hogy $O(n^2)$ -es. Válasszuk ki, hogy melyik ez a lépésszám, valamint megfelelő c konstans és n_0 küszöbérték megadásával bizonyítsuk is ezt be.

$\log_2(n^2) + \frac{n(n-1)(n-2)}{2010}$, $2020n \log_2 n + \frac{2^n \cdot n(n-1)}{3^n}$, $20n^2(\log_2 n)^2 - 8$

5. Alkalmas c konstans és n_0 küszöbérték megadásával bizonyítsuk be, hogy egy $\binom{n}{5}$ lépésszámú algoritmus $O(n^5)$ -es. (Emlékeztetőül: $\binom{n}{k} = \frac{n!}{k!(n-k)!}$.)
6. Mutassuk meg, hogy az alábbi algoritmusok lépésszáma $O(n^2)$ -es. (Lépésnek az értékadásokat, az összeadásokat, egy $*$ kiírását, a párosság eldöntését, illetve az eredmény visszaadását tekintjük.)

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

ciklus: i fut 0-tól $(n-1)$ -ig

$j := 0$

ciklus: amíg $j < i$

print $"*"$

$j := j + 1$

ciklus vége

ciklus vége

(b) **Input:** két $n \geq 1$ darab egész számot tartalmazó A és B tömb

ciklus: i fut 0-tól $(n-1)$ -ig

ha $A[i]$ páros:

ciklus: j fut 0-tól $(n-1)$ -ig

$B[j] := B[j] + 17$

ciklus vége

ciklus vége

return B

7. Az alábbi algoritmusban lépésnek az értékadásokat, az összeadásokat és az eredmény visszaadását tekintjük.

(a) Mutassuk meg, hogy az algoritmus lépésszáma $O(n)$.

(b) Mutassuk meg, hogy az algoritmus lépésszáma $O(n^2)$.

Input: egy $n \geq 2$ darab számot tartalmazó A tömb

ciklus: i fut 0-tól $(n - 1)$ -ig

szamlalo := -17

ciklus: j fut 0-tól 2025-ig

$A[i] := A[i] + A[j]$

szamlalo := szamlalo + 2

ciklus vége

ciklus vége

return A

8. Mutassuk meg, hogy az alábbi algoritmusok lépésszáma $O(n^2)$ -es. (Lépésnek az értékadásokat, az összehasonlításokat, az összeadásokat, a kivonásokat, illetve az eredmény visszaadását tekintjük.)

(a) **Input:** egy $n \geq 1$ darab számot tartalmazó A tömb

$S := 0$

ciklus: i fut 0-tól $(n - 1)$ -ig

$j := i$

$M := 0$

ciklus: amíg $j \geq 0$

$M := M + A[j]$

$j := j - 1$

ciklus vége

$S := S + M$

ciklus vége

return S

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

kicsi := 1

nagy := n

szamlalo := 0

ciklus: amíg kicsi < nagy

ciklus: i fut 0-tól n -ig

szamlalo := szamlalo + 1

ciklus vége

kicsi := kicsi + 1

nagy := nagy - 1

ciklus vége

return szamlalo

9. Mutassuk meg, hogy az alábbi algoritmus lépésszáma $O(n^3)$ -ös. (Lépésnek az értékadásokat, az összehasonlításokat, illetve az eredmény visszaadását tekintjük. Az $(n \times n)$ -es A és B táblázatok i -edik sorának j -edik elemét $A[i, j]$, illetve $B[i, j]$ jelöli tetszőleges $i \in \{0, 1, \dots, n - 1\}$ esetén.)

Input: két darab, egész számokat tartalmazó $(n \times n)$ -es táblázat A és B

ciklus: i fut 0-tól $(n - 1)$ -ig

ciklus: j fut 0-tól $(n - 1)$ -ig

ciklus: k fut 0-tól $(n - 1)$ -ig

ha $A[i, k] == A[j, k]$:

$B[i, j] := 1$

ciklus vége

ciklus vége

ciklus vége

return B