1. Egy algoritmus lépésszáma $100 \cdot n^2 + 8 \log_2 n + 17$. Alkalmas c konstans és n_0 küszöbérték megadásával bizonyítsuk be, hogy ez az algoritmus (a) $O(n^2)$-es; (b) $O(n^3)$-ös.

2. Jobbra látható egy algoritmus, melynek bemenete egy $n \geq 2$ egész szám. Egy lépésnek egy $*$ kiírását tekintjük. Mutassuk meg, hogy az algoritmus lépésszáma (a) $O(n^2)$; (b) $O(2^n)$.

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

---

3. Egy algoritmus lépésszáma $10 \cdot n^3 + 3 \cdot \sqrt{n} - 5n - 2$. Alkalmas c konstans és n_0 küszöbérték megadásával bizonyítsuk be, hogy ez egy $O(n^3)$-ös algoritmus.

4. Alább látható két algoritmus pszeudokódja. Mindkettő bemenete egy-egy $n \geq 2$ darab egész számot tartalmazó T tömb. Lépésnek az összehasonlításokat, értékadásokat és a cseréket tekintjük.
   (a) Mutassuk meg, hogy a baloldali algoritmus lépésszáma $O(n)$.
   (b) Mutassuk meg, hogy a jobboldali algoritmus lépésszáma $O(n^2)$.
   (c*) Mit csinálnak az algoritmusok?

ciklus: i **fut** 0**-tól** $(n-2)$**-ig**
  ha $T[i] > T[i+1]$**:**
    csere$\big(T[i], T[i+1]\big)$
ciklus vége

ciklus: j **fut** $(n-1)$**-től** 1**-ig**
  ciklus: i **fut** 0**-tól** $(j-1)$**-ig**
    ha $T[i] > T[i+1]$**:**
      csere$\big(T[i], T[i+1]\big)$
  ciklus vége
ciklus vége

5. Jobbra látható egy algoritmus lépésszáma. Mutassuk meg, hogy ez az algoritmus (a) $O(n)$-es; (b) $O(n^2)$-es.

$$L(n) = \begin{cases} 8 \cdot n^3, & \text{ha } n < 6, \\ 3 \cdot n, & \text{különben.} \end{cases}$$

6.* Legyenek f_1 és f_2 pozitív értékkészletű függvények. Bizonyítsuk be, hogy
   (a) $O(f_1) + O(f_2) = O\big(\max(f_1, f_2)\big)$; (b) $O(f_1) \cdot O(f_2) = O(f_1 \cdot f_2)$.

---

7. Egy algoritmus lépésszáma $10^{10} \cdot n + 10 \cdot n \cdot \log_2 n + 17n^3$. Alkalmas c konstans és n_0 küszöbérték megadásával bizonyítsuk be, hogy ez az algoritmus (a) $O(n^3)$-ös, (b) $O(n^5)$-es.

8. Jobbra látható egy algoritmus pszeudokódja, melynek bemenete két $n \geq 1$ darab egész számot tartalmazó A és B tömb. Mutassuk meg, hogy az algoritmus lépésszáma $O(n^2)$-es. (Lépésnek az értékadásokat és az összehasonlításokat tekintjük.)

ciklus: i **fut** 0**-tól** $(n-1)$**-ig**
  ha $A[i]$ páros**:**
    ciklus: j **fut** 0**-tól** $(n-1)$**-ig**
      $B[j] \coloneqq B[j] + 17$
    ciklus vége
ciklus vége
return B

9.* Alkalmas c konstans és n_0 küszöbérték megadásával bizonyítsuk be, hogy
   (a) egy $\binom{n}{5}$ lépésszámú algoritmus $O(n^5)$-es; (b) egy $1 + 2 + 4 + \ldots + 2^n$ lépésszámú algoritmus $O(2^n)$-es.

10.* Jobbra látható az **algoritmus** nevű eljárás pszeudokódja, melynek a bemenete egy $n = 2^k$ darab egész számot tartalmazó T tömb.
   (a) Mutassuk meg, hogy az algoritmus lépésszáma $O(n)$-es. (Lépésnek az összehasonlításokat, az értékadásokat és az eredmény visszaadását tekintjük.)
   (b) Mit csinál az algoritmus?

ha $n == 1$**:**
  return $T[0]$
különben:
  `egyik` $\coloneqq$ **algoritmus**$(T[0 : n/2 - 1])$
  `masik` $\coloneqq$ **algoritmus**$(T[n/2 : n - 1])$
  ha `egyik` $\leq$ `masik`**:**
    return `egyik`
  return `masik`

11.* Jobbra látható az **algoritmus** nevű eljárás pszeudokódja, melynek a bemenete egy $n = 2^k - 1$ darab egész számot tartalmazó, növekvően rendezett T tömb.
   (a) Mutassuk meg, hogy az algoritmus lépésszáma $O(\log_2 n)$-es. (Lépésnek az összehasonlításokat és az eredmény visszaadását tekintjük.)
   (b) Mit csinál az algoritmus?

ha $n == 1$**:**
  return $T[0] == 7$
különben:
  ha $T[(n-1)/2] < 7$**:**
    return algoritmus$(T[0 : (n-3)/2])$
  különben ha $T[(n-1)/2] > 7$**:**
    return algoritmus$(T[(n+1)/2 : n-1])$
  különben:
    return `true`