1. Adott a $T = [3, 5, 1, 6]$ tömb.
   (a) Futtassuk az órán tanult minimumkereső algoritmust T-re.
   (b) Futtassuk az órán tanult kiválasztásos rendezést T-re.

2. Adott egy n darab különböző számot tartalmazó T tömb, és egy m darab különböző számot tartalmazó S tömb. Adjunk algoritmust annak eldöntésére, hogy a T tömb legkisebb eleme egyenlő-e az S tömb legkisebb elemével. (Lépésnek az értékadásokat, az összehasonlításokat és az eredmény visszaadását tekintjük.)
   (a) Írjuk le az algoritmust (pszeudokóddal vagy szöveggel).
   (b) Magyarázzuk el, hogy az algoritmus miért helyes.
   (c) Határozzuk meg az algoritmus lépésszámát, és indokoljuk is az eredményt.

---

3. Futtassuk az órán tanult kiválasztásos rendezést a $T = [5, 2, 9, 7]$ tömbre. Hány összehasonlítás történik a rendezés teljes futása során?

4. Adjunk algoritmust egy n darab különböző számot tartalmazó tömb legnagyobb elemének megkeresésére. (Lépésnek az összehasonlításokat, az értékadásokat és az eredmény visszaadását tekintjük.)
   (a) Írjuk le az algoritmust (pszeudokóddal vagy szöveggel).
   (b) Magyarázzuk el, hogy az algoritmus miért helyes.
   (c) Határozzuk meg az algoritmus lépésszámát, és indokoljuk is az eredményt.

5. A $T = [5, x, 9, 4, 1]$ tömbön, ami csupa különböző egész számot tartalmaz, futtattuk a kiválasztásos rendezést. Az algoritmus futtatása közben valamikor az $[1, 4, 9, x, 5]$ közbülső állapotot láttuk. Határozzuk meg az x egész szám összes lehetséges értékét (és indokoljuk meg, hogy x értéke nem lehet más).

6. Adott egy n darab számot tartalmazó T tömb, és egy m darab számot tartalmazó S tömb. Adjunk algoritmust annak eldöntésére, hogy a T tömb minden eleme kisebb-e, mint az S tömb minden eleme. (Lépésnek az értékadásokat, az összehasonlításokat és az eredmény visszaadását tekintjük.)
   (a) Írjuk le az algoritmust (pszeudokóddal vagy szöveggel).
   (b) Magyarázzuk el, hogy az algoritmus miért helyes.
   (c) Határozzuk meg az algoritmus lépésszámát, és indokoljuk is az eredményt.

7. Van egy számítógépes programunk, ami egy 100 darab számot tartalmazó tömbön a jelenlegi gépünkön 1 nap alatt fut le. Beszereztünk egy 16-szor gyorsabb számítógépet. Ugyanaz a program mekkora bemeneten fut le az új gépen egy nap alatt, ha a program lépésszáma n darab számot tartalmazó tömb esetén (a) n; (b) $3n$; (c) n^2?

8.* Van egy számítógépes programunk, ami egy k darab számot tartalmazó tömbön a jelenlegi gépünkön 1 nap alatt fut le. Beszereztünk egy 16-szor gyorsabb számítógépet. Ugyanaz a program mekkora bemeneten fut le az új gépen egy nap alatt, ha a program lépésszáma n darab számot tartalmazó tömb esetén (a) n; (b) n^2; (c) 2^n?

---

9. A $T = [5, x, 9, 4, 1]$ tömbön, ami csupa különböző egész számot tartalmaz, futtattuk a kiválasztásos rendezést. Az algoritmus futtatása közben valamikor az $[1, x, 4, 9, 5]$ közbülső állapotot láttuk. Határozzuk meg az x egész szám összes lehetséges értékét (és indokoljuk meg, hogy x értéke nem lehet más).

10. Adott egy n darab számot tartalmazó T tömb. Adjunk algoritmust annak eldöntésére, hogy van-e olyan szám, ami többször is előfordul T-ben. (Lépésnek az értékadásokat, az összehasonlításokat és az eredmény visszaadását tekintjük.)
    (a) Írjuk le az algoritmust (pszeudokóddal vagy szöveggel).
    (b) Magyarázzuk el, hogy az algoritmus miért helyes.
    (c) Határozzuk meg az algoritmus lépésszámát, és indokoljuk is az eredményt.

11.* 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.

12.* 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
(a) $L(n) = 2 + L(n-1)$; (b) $L(n) = 2 \cdot L(n-1)$; (c)** $L(n) = 2 \cdot L(n-1) + 2$.