1. Az alábbi függvények közül melyikre igaz, hogy $O(n^2)$?
   $f_1(n) = 11n^2 + 100000$ $f_2(n) = 8n^2 \log_2 n$ $f_3(n) = 1{,}5n + 3\sqrt{n}$

2. Az alábbi pszeudokódban egy $*$ kiírása számít lépésnek. Mutassa meg, hogy a kód lépésszáma $O(n^3)$.

```
for i = 0 to n-1:
    for j = i+1 to n:
        print  j  darab *
```

3. (a) Lássa be, hogy a buborékrendezés lépésszáma $O(n^2)$ (ezt a rendező algoritmust Prog1-ből már tanulták, de itt láthatják a pszeudokódját ismét). Releváns lépésnek az összehasonlítás és a csere számít. Igaz-e, hogy az algoritmus lépésszáma $O(n^3)$?
   (b) Lássa be, hogy a buborékrendezés lépésszáma $\Omega(n^2)$.
   (c) Igaz-e, hogy a buborékrendezés lépésszáma $\Theta(n^2)$?

```
ciklus i = n-tól 2-ig:
    ciklus j = 1-tól (i-1)-ig:
        ha   A[j] > A[j+1]:
            csere A[j] és A[j+1]
    ciklus vége
ciklus vége
```

---

4. Az alábbi függvényeket rendezze nagyságrend szerint nem csökkenő sorozatba: ha f_i után közvetlenül f_j következik a sorban, akkor $f_i(n) \in O(f_j(n))$ teljesüljön!
   $f_1(n) = 8n^3$ $f_2(n) = 5\sqrt{n} + 1000n$ $f_3(n) = 2^{(\log_2 n)^2}$ $f_4(n) = 1514n^2 \log_2 n$

5. Az $A[1 : n]$ tömb (bináris keresést használó) beszúrásos rendezése $n - 1$ körből áll: az i. körben $(1 \leq i \leq n-1)$ a már rendezett $A[1 : i]$ tömbben megkeressük az $A[i+1]$ elem helyét bináris kereséssel, majd az $A[i+1]$ elemet szomszédos elemek cseréjével balra mozgatjuk addig, amíg a megtalált pozícióba nem érkezik. (a) Lássa be, hogy az algoritmus lépésszáma $O(n^2)$, releváns lépésnek az összehasonlítás és a csere számít. (b) Lássa be, hogy az algoritmus lépésszáma $\Omega(n^2)$ és $\Theta(n^2)$ is.

6. Tekintsük az $f_1(n) = 1{,}5\,n!$ és $f_2(n) = 200\,(n-1)!$ függvényeket. Melyik igaz és melyik nem az alábbiak közül?
   $f_1 \in O(f_2)$ $f_2 \in O(f_1)$ $f_1 \in \Omega(f_2)$ $f_2 \in \Omega(f_1)$ $f_1 \in \Theta(f_2)$ $f_2 \in \Theta(f_1)$

7. Adjon O becslést a következő függvényekre:
   $(n^2 + 8)(n + 1)$ $(n \log n + n^2)(n^3 + 2)$ $(n! + 2^n)(n^3 + \log(n^2 + 1))$ $(2^n + n^2)(n^3 + 3^n)$

---

8. Ugyanarra a feladatra van két algoritmusunk A és B. A maximális lépésszámot leíró függvényeket jelölje f_A és f_B. Tudjuk, hogy $f_A(n) \in O(f_B(n))$. Következik-e ebből, hogy
   a) A minden bemeneten gyorsabb, mint B? b) A véges sok bemenet kivételével gyorsabb, mint B?
   c) A megfelelően nagy bemenetekre gyorsabb, mint B?

9. Bizonyítsa be, hogy ha f_1, f_2, g_1, g_2 pozitív értékkészletű függvények és $f_1(n) \in O(g_1(n))$ és $f_2(n) \in O(g_2(n))$, akkor
   (a) $f_1(n) + f_2(n) \in O(\max(g_1(n), g_2(n))$ (b) $f_1(n) \cdot f_2(n) \in O(g_1(n) \cdot g_2(n))$.

10. Mely $a, b > 1$ egész számokra teljesülnek az alábbiak?
    $n^a \in O(n^b)$ $2^{an} \in O(2^{bn})$ $\log_a n \in O(\log_b n)$

11. Tegyük fel, hogy n 2-hatvány. Bizonyítsuk be, hogy ahhoz, hogy n különböző számból a két legnagyobb elemet kiválasszuk, $n + \log_2 n - 2$ összehasonlítás elégséges.

12. Az $A[1 : n]$ tömb piros és zöld elemeket tartalmaz. Szeretnénk átrendezni úgy, hogy az egyszínű elemek folytonosan helyezkedjenek el (elöl az összes piros, utána a zöldek vagy fordítva). Egy megengedett *lépés* két szomszédos tömbelem cseréje. Javasoljunk konstans szorzó erejéig optimális lépésszámú algoritmust.

13. Adott n chip, melyek képesek egymás tesztelésére a következő módon: ha összekapcsolunk két chipet, mindkét chip nyilatkozik a másikról, hogy hibásnak találta-e. Egy hibátlan chip korrektül felismeri, hogy a másik hibás-e, míg egy hibás chip akármilyen választ adhat. Tegyük fel, hogy a chipek több, mint a fele korrekt. Adjunk algoritmust, mely n-nél kevesebb fenti tesztet használva kikeres egy jó chipet.