Függvények nagyságrendje

Algoritmuselmélet

1. gyakorlat

2023.

Nagyságrendek: O, Ω, Θ.

Ha $f, g: \mathbb{N} \to \mathbb{R}^+$ függvények, akkor

(i) $f \in O(g)$ jelöli azt a tényt, hogy léteznek olyan $c, n_0 > 0$ állandók, hogy minden $n \geq n_0$ esetén $f(n) \leq c \cdot g(n)$ teljesül;

(ii) $f \in \Omega(g)$ jelöli azt a tényt, hogy léteznek olyan $c, n_0 > 0$ állandók, hogy minden $n \geq n_0$ esetén $f(n) \geq c \cdot g(n)$ teljesül;

(iii) $f \in \Theta(g)$ jelöli azt a tényt, hogy $f \in O(g)$ és $f \in \Omega(g)$ is teljesül.

1. Bizonyítsuk be, hogy

   (a) $n^2 - 4n + 7 \in \Theta(n^2)$;

   (b) $100(n-1)! \in O(n!)$, de $100(n-1)! \notin \Omega(n!)$;

   (c) $\log_{10} n \in \Theta(\log_2 n)$;

   (d) $2^n \in O(3^n)$, de $2^n \notin \Omega(3^n)$;

   (e) $2^{n+1} \in \Theta(2^n)$, de $2^{2n} \notin O(2^n)$;

   (f) $1 + 2 + \ldots + n \in \Theta(n^2)$;

   (g) $1 + 2 + 2^2 + \ldots + 2^n \in \Theta(2^n)$;

   (h) $\sqrt{2n^2 + 3n + 15} \in O(n)$.

2. Adjunk minél jobb O becslést a következő függvényekre.

   (a) $(n^2 + 8)(n + 1)$

   (b) $(n \log_2 n + n^2)(n^3 + 2)$

   (c) $5^n + n^n + n!$

   (d) $(n! + 2^n)\big(n^3 + \log_2(n^2 + 1)\big)$

   (e) $(2^n + n^2)(n^3 + 3^n)$

3. Az alábbi függvényeket rendezzük olyan sorozatba, hogy ha f_i után közvetlenül f_j következik a sorban, akkor $f_i(n) \in O\big(f_j(n)\big)$ teljesüljön.

$$f_1(n) = 2^{(\log_2 n)^2}, \qquad f_2(n) = 8^{\log_2 n}, \qquad f_3(n) = 1514 n^2 \log_2 n$$

4. Mely $a, b > 1$ számokra teljesülnek az alábbiak?

   (a) $n^a \in O(n^b)$

   (b) $2^{an} \in O(2^{bn})$

   (c) $\log_a n \in O(\log_b n)$

5. Legyenek f és g pozitív értékkészletű függvények. Bizonyítsuk be, hogy ha $f \in O(g)$ fennáll, akkor $g \in \Omega(f)$ is teljesül.

6. Legyenek f_1, f_2, g_1, g_2 pozitív értékkészletű függvények. Bizonyítsuk be, hogy ha $f_1 \in O(g_1)$ és $f_2 \in O(g_2)$ fennáll, akkor az alábbiak is teljesülnek.

   (a) $f_1 + f_2 \in O\big(\max(g_1, g_2)\big)$

   (b) $f_1 \cdot f_2 \in O(g_1 \cdot g_2)$

7. Ugyanarra a feladatra van két algoritmusunk, A és B. A maximális lépésszámot leíró függvényeket jelölje f_A és f_B. Tudjuk, hogy $f_A(n) \in O\big(f_B(n)\big)$. 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?

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

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

9. (a) Lássuk be, hogy a buborékrendezés lépésszáma $O(n^2)$ (az algoritmus pszeudokódja alább olvasható); releváns lépésnek az összehasonlítás és a csere számít.

   (b) Igaz-e, hogy a buborékrendezés lépésszáma $O(n^3)$?

   (c) Lássuk be, hogy a buborékrendezés lépésszáma $\Omega(n^2)$.

   (d) Igaz-e, hogy a buborékrendezés lépésszáma $\Theta(n^2)$?

```
for i = n to 2:
    for j = 1 to i-1:
        if A[j] > A[j+1]:
            csere A[j] és A[j+1]
```

10. Az $A[1:n]$ tömb (bináris keresést használó) beszúrásos rendezése $n-1$ körből áll: az i-edik körben ($1 \le i \le n-1$) 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) Lássuk be, hogy az algoritmus lépésszáma $O(n^2)$; releváns lépésnek az összehasonlítás és a csere számít.

    (b) Lássuk be, hogy az algoritmus lépésszáma $\Omega(n^2)$ és $\Theta(n^2)$ is.

11. Jelölje egy algoritmus maximális lépésszámát az n méretű bemeneteken $L(n)$. Adjunk felső becslést az $L(n)$ nagyságrendjére, ha tudjuk, hogy $L(1) = 2$ és tetszőleges $n > 1$ esetén

    (a) $L(n) = L(n-1) + 3$;

    (b) $L(n) = L(n-1) + 3n$;

    (c) $L(n) = L\big(\lceil n/2 \rceil\big) + 3$;

    (d) $L(n) = 4L\big(\lceil n/2 \rceil\big) + 3$.

    A (c) és (d) eseteket elegendő 2-hatványokra meggondolni.

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

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

14. 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 hibátlan. Adjunk algoritmust, mely n-nél kevesebb fenti tesztet használva kikeres egy hibátlan chipet.