

Oszthatóság, prímszámok

Feladatsor

Diszkrét modellek alkalmazásai

Matematikai kérdések

1. Írjuk fel a maradékos osztás tételét. Mutassuk be 21-re és 5-re.
2. Definiáljuk az osztó fogalmát. Mutassuk be a definíciót a 12 egy tetszőleges osztójára.
3. Soroljuk fel 16 pozitív osztóit. Ezek közül melyek valódi osztók, ill. nemtriviális osztók?
4. Definiáljuk az asszociált fogalmát. Mik 10 asszociáltjai?
5. Definiáljuk a prímszám fogalmát. Soroljuk fel 15-ig az összes prímszámot.
6. Hogy nevezzük az olyan számot, amelynek kettőnél több pozitív osztója van? Írjunk egy példát.
7. Mit mond a számelmélet alaptétele?
8. Számítsuk ki a következő számok prímtényezős felbontását (kanonikus alakban):
a) 30, b) 36, c) 37, d) 64, e) 150, f) 200, g) 3000.
9. Mit jelent $\tau(n)$ és $\sigma(n)$? Mennyi $\tau(12)$ és $\sigma(12)$?
10. Számítsuk ki – kétféleképpen – a következő értékeket:
a) $\tau(7)$, b) $\tau(8)$, c) $\tau(9)$, d) $\tau(10)$, e) $\tau(12)$, f) $\tau(24)$, g) $\tau(32)$, h) $\tau(300)$.
11. Mit jelent a tökéletes szám? Tökéletes szám-e a 28?

Programozási feladatok

12. A Sage prímszámos függvényei segítségével oldjuk meg az alábbi feladatokat – mindegyiket legalább háromféleképpen.
- a) Írjuk ki 1-től 100-ig az összes prímszámot.
 - b) Írjuk ki az első 100 prímszámot.
 - c) Hány prímszám van 1000-ig?
13. Írjunk függvényt, amely eldönti, hogy egy adott $n \in \mathbb{N}$ szám prímszám-e (a Sage prímszámos függvényei nélkül). Ellenőrizzük a megoldást az `is_prime()` függvénnyel 0-tól 100-ig, valamint futtassuk le az alábbi prímszámokra is: 1237, 1212121, 1234567891.
14. Írjunk függvényt, amely visszaadja egy szám pozitív osztóit egy listában (a Sage matematikai függvényei nélkül). Ellenőrizzük a megoldást a `divisors()` függvénnyel 1-től 100-ig, valamint futtassuk le az alábbi számokra is: 1000, 1111111, 2^{30} .
15. Ellenőrizzük 1-től 100-ig, hogy az oszthatóság tranzitív.
16. Hozzunk létre egy listát a 600 alatti ikerprímpárokkal: [(3, 5), (5, 7), (11, 13), ...].
17. Írjunk függvényt, amely adott n -ig előállítja az összes prímszámot Eratoszthenész szitájával. (Ne használjuk a beépített prímszámos függvényeket.) A függvény a prímszámok listáját adja vissza (amit az algoritmus `bool`-listája alapján hozunk létre a végén). Hívjuk meg $n = 100$ -ra, és írassuk ki a visszaadott listát. Hívjuk meg $n = 100000$ -re is, és ellenőrizzük, hogy a lista utolsó prímszáma helyes-e (a `previous_prime()`-mal).
18. a) Írjunk függvényt, amely kiszámítja $\tau(n)$ -et a definíció alapján. Ellenőrzéshez használjuk a `number_of_divisors()` függvényt.
b) Írjunk függvényt, amely kiszámítja $\tau(n)$ -et a prímtényezős felbontás alapján.
c) Írassuk ki 100-ig azokat az n -eket, amelyekre $\tau(n)$ páratlan. Mi a szabály? Vajon miért?
19. Keressük meg az összes tökéletes számot 10000-ig.