Euler–Fermat-tétel és RSA

Feladatsor

Diszkrét modellek alkalmazásai

Matematikai kérdések

1. Mit jelent a redukált maradékrendszer? Írjunk rá *két* példát modulo 6.

2. Definiáljuk $\varphi(n)$-et. Mutassuk be a definíciót $\varphi(6)$-ra.

3. Számítsuk ki $\varphi()$ következő értékeit:
a) $\varphi(25)$; b) $\varphi(27)$; c) $\varphi(30)$; d) $\varphi(36)$; e) $\varphi(45)$; f) $\varphi(81)$; g) $\varphi(100)$; h) $\varphi(600)$.

4. Írjuk fel az Euler–Fermat-tételt. Adjunk rá *két* példát modulo 6.

5. Számítsuk ki a következő hatványokat az Euler–Fermat-tétel segítségével:
a) $7^5 \bmod 6$; b) $137^{138} \bmod 10$; c) $52^{45} \bmod 25$; d) $321^{321} \bmod 100$; e) $1231^{642} \bmod 600$.

6. a) Hogyan generálja az RSA az n modulust?
b) Hogyan generálja az RSA a d titkos kitevőt?
c) Hogyan titkosít az RSA egy m üzenetet?
d) Hogyan fejt meg az RSA egy c titkosított kódot?
e) A következő RSA paraméterek közül melyek nyilvánosak: d, e, n, p, q?

Programozási feladatok

7. Írjunk függvényt, amely visszaadja az invertálható maradékok listáját egy adott $m \in \mathbb{N}^+$ modulusra. Írassuk ki ezeket a listákat $m = 1$-től 20-ig. Mi lesz a listák hossza?

8. a) Írjunk függvényt, amely kiszámítja $\varphi(n)$-et a definíció alapján.
b) Írjunk függvényt, amely kiszámítja $\varphi(n)$-et a prímtényezős felbontás alapján.
c) Teszteljük mindkét megoldást $n = 1$-től 100-ig az `euler_phi()` beépített függvénnyel.
d) $\varphi(n)$-ről bebizonyítható, hogy minden $n \geq 3$-ra páros. Ellenőrizzük ezt $n = 1000$-ig.

9. Mennyi $\sum_{d|n} \varphi(d)$, azaz $\varphi(d)$-k összege n minden d pozitív osztójára? Számítsuk ki az összeget különböző n-ekre, fogalmazzunk meg egy sejtést, és ellenőrizzük azt minden n-re 1-től 1000-ig.

10. a) Generáljunk egy RSA-kulcspárt. Az egyszerűség kedvéért használhatjuk a (nem biztonságos) `random_prime(L)`-et, amely visszaad egy L-nél kisebb véletlen prímszámot. Először próbáljuk ki $L = 1000$-re, és ha működik, akkor végül írjuk át $L = 2^{120}$-ra. A nyilvános kitevő legyen az első olyan páratlan szám, amely megfelel az RSA feltételeinek.
b) Titkosítsunk egy tetszőleges m üzenetet ($2 \leq m < n$) a nyilvános kulcs segítségével.
c) Állítsuk vissza a titkosított c értékből az m üzenetet a titkos kulcs segítségével.

(Ne használjunk olyan műveletet, amely az RSA feltörését igényelné, pl. n faktorizálását.)

11. Gyorsítsuk az RSA implementációját a kínai maradéktétel segítségével.

12. a) Írjunk függvényt, amely a Fermat-prímteszt segítségével eldönti, hogy egy adott $n \in \mathbb{N}^+$ szám prímszám-e. (Egyetlen véletlen a-t használjunk.)
b) Keressünk olyan n-et, amelyre a teszt hibás eredményt ad.
c) Javítsuk a tesztet úgy, hogy megismételjük k-szor (külön-külön véletlen a-kra), ahol k szintén bemenő paraméter.

13. Keressük meg 1-től 10000-ig az $a = 2$ alaphoz tartozó összes Fermat-álprímet.

14. Keressük meg 1-től 10000-ig az összes Carmichael-számot.

15. Implementáljuk a Miller–Rabin-prímtesztet.

16. Keressük meg 1-től 10000-ig az $a = 2$ alaphoz tartozó összes erős álprímet.