Számelméleti algoritmusok

Bevezetés a számításelméletbe 1
3. gyakorlat

Áruló

Legyenek $a < m$ pozitív egészek. Az a számot az m árulójának nevezzük, ha $(a, m) = 1$ és $a^{\varphi(m)} \not\equiv 1 \pmod{n}$

Carmichael-szám

Egy m pozitív egész számot Carmichael-számnak nevezzük, ha összetett és nincs árulója.

1. Határozzuk meg
   (a) 899 és 493;
   (b) 346 és 158;
   (c) 24961 és 9483

   legnagyobb közös osztóját.
2. Milyen maradékot ad
   (a) 3^{45} 79-cel osztva;
   (b) 3^{169} 91-gyel osztva;
   (c) 5^{300} 623-mal osztva;
   (d) 5^{85} 155-tel osztva?
3. Határozzuk meg a $10x \equiv 24 \pmod{m}$ lineáris kongruencia megoldásait modulo m, ahol
   (a) $m = 15$, illetve
   (b) $m = 16$.
4. Hány olyan 2012-nél kisebb pozitív egész szám van, amely 19-cel osztva 10 maradékot ad és 37-tel osztva 15 maradékot ad?
5. Legyen $n = 20181210$. Az előadáson tanult megfelelő algoritmus alkalmazásával határozzuk meg $45n + 12$ és $35n + 9$ legnagyobb közös osztóját.
6. Legyen $n = 123456$. Az előadáson tanult megfelelő algoritmus alkalmazásával határozzuk meg $12n + 6$ és $9n + 4$ legnagyobb közös osztóját.
7. Az alábbi két C kód mindegyike a bemenetként (10-es számrendszerben) kapott $a, b > 0$ egészek összegét számítja ki (persze feleslegesen bonyolultan). Tegyük fel, hogy a kódok végrehajtásakor a gép az alapműveleteket az (alsó tagozatban tanult) „írásbeli” összeadás, szorzás, stb. segítségével végzi el. Döntsük el, hogy az eljárások polinomiálisak-e. (A `ceil(b/2.0)` a $b/2$ felső egészrészét, míg `floor(b/2.0)` a $b/2$ alsó egészrészét adja vissza.)

(a)
```
while (b > 0) {
   a = a+1;
   b = b-1;
}
printf("Összeg: %d", a);
```

(b)
```
while (b > 0) {
   a = a + ceil(b/2.0);
   b = floor(b/2.0);
}
printf("Összeg: %d", a);
```

8. Az alábbi C kódok közül az első $\lfloor\sqrt{n}\rfloor$-t, a második $\lfloor\log_2 n\rfloor$-t számítja ki bármely bemenetként (10-es számrendszerben) kapott $n > 0$ egész esetén. Tegyük fel, hogy a kódok végrehajtásakor a gép az (alsó tagozatban tanult) „írásbeli" összeadás, szorzás, stb. segítségével végzi el. Döntsük el, hogy az eljárások polinomiálisak-e.

(a)
```
x = 0; y = 0;
while (y <= n) {
   x = x+1;
   y = x*x;
}
printf("Eredmény: %d", x-1);
```

(b)
```
x = 0; y = 1;
while (y <= n) {
   x = x+1;
   y = 2*y;
}
printf("Eredmény: %d", x-1);
```

9. Az alábbi C kód a bemenetként (10-es számrendszerben) kapott n pozitív egész négyzetét számítja ki. Tegyük fel, hogy a kód végrehajtásakor a gép az alapműveleteket az „írásbeli" összeadás és kivonás segítségével végzi el. Döntsük el, hogy az eljárás polinomiális-e.

```
x = n; y = 0;
while (x > 0) {
   x = x-1;
   y = y+n;
}
printf("Eredmény: %d", y);
```

10. Az alábbi C kód a bemenetként (10-es számrendszerben) kapott $0 < a < n$ egészek esetén az n-nek az a-nál nemnagyobb osztói közül a legnagyobbat számítja ki. Tegyük fel, hogy a kód végrehajtásakor a gép az (alsó tagozatban tanult) „írásbeli" összeadás, szorzás, stb. segítségével végzi el. Döntsük el, hogy az eljárás polinomiális-e.

```
while (n%a != 0) {
   a = a-1;
}
printf("Eredmény: %d", a);
```

11. Mutassuk meg, hogy az 561 Carmichael-szám.

12. Létezik-e páros Carmichael-szám?

13. Legyen n egy 8-cal osztható, de 3-mal nem osztható pozitív egész szám. Mutassuk meg, hogy a 3 árulója n-nek.