

Pollard- ρ -Algorithmus für diskrete Logarithmierung

In dem Pollard- ρ -Algorithmus suchten wir eine Wiederholung in der Reihe der Zahlen b_0, b_1, \dots . Der Algorithmus kann wesentlich verbessert werden, wenn man dafür den Floyd-Algorithmus benutzt. Der Floyd-Algorithmus wird nach Aufgabe 1 verständlich sein.

Aufgabe 1. Sei S eine endliche Menge der natürlichen Zahlen und sei $f : S \mapsto S$ eine Abbildung. Sei b_0, b_1, \dots , eine unendliche Reihe der Elemente von S , so daß $b_{i+1} = f(b_i)$ für $i \geq 0$ ist. Da S eine endliche Menge ist, wiederholt sich die Reihe ab einer bestimmten Stelle, d.h. es existieren verschiedene Indexe i, j mit $b_i = b_j$. Beweisen Sie, daß ein Index m existiert mit $b_m = b_{2m}$.

Floyd-Algorithmus für die Suche nach einer Wiederholung in der Reihe b_0, b_1, \dots

1. Berechnen wir das Paar (b_0, b_1) .
2. Wenn ein Paar (b_i, b_{2i}) bekannt ist und $b_i \neq b_{2i}$ ist, berechnen wir das Paar $(b_{i+1}, b_{2(i+1)})$.
3. Wir stoppen nur dann, wenn $b_m = b_{2m}$ für einen m ist.

Der Algorithmus fordert nicht viel Platz auf der Speicherplatte: in jedem Moment müssen wir nur ein Paar der Zahlen speichern.

Anmerkung für Aufgabe 2. Sei wollen wir x aus der folgenden Kongruenz finden:

$$a \equiv g^x \pmod{q}.$$

In dem standarden Pollard- ρ -Algorithmus haben wir eine Reihe der Tripel mit folgendem Gesetz¹ definiert: $(b_0, y_0, z_0) = (a, 0, 1)$,

mod 3-Gesetz:

$$(b_{i+1}, y_{i+1}, z_{i+1}) = \begin{cases} (b_i^2, 2y_i, 2z_i) & \text{falls } b_i \equiv 0 \pmod{3} \text{ ist,} \\ (b_i a, y_i, z_i + 1) & \text{falls } b_i \equiv 1 \pmod{3} \text{ ist,} \\ (b_i g, y_i + 1, z_i) & \text{falls } b_i \equiv 2 \pmod{3} \text{ ist.} \end{cases}$$

Wir können aber andere Gesetze benutzen:

Würfel-Gesetz. Erst konstruieren wir eine Reihe der zufälligen (unabhängigen) Zahlen: x_1, x_2, \dots , wobei $x_i \in \{0, 1, 2\}$ für alle i ist. Dann setzen wir

$$(b_{i+1}, y_{i+1}, z_{i+1}) = \begin{cases} (b_i^2, 2y_i, 2z_i) & \text{falls } x_i = 0 \text{ ist,} \\ (b_i a, y_i, z_i + 1) & \text{falls } x_i = 1 \text{ ist,} \\ (b_i g, y_i + 1, z_i) & \text{falls } x_i = 2 \text{ ist.} \end{cases}$$

¹**Wichtig:** die Zahlen b_i werden modulo q berechnet und die Zahlen y_i und z_i werden modulo $q - 1$ berechnet.

3-Stück-Gesetz. Bezeichnen wir $c = \lfloor \frac{q-1}{3} \rfloor$.

$$(b_{i+1}, y_{i+1}, z_{i+1}) = \begin{cases} (b_i^2, 2y_i, 2z_i) & \text{falls } b_i \in \{1, \dots, c\} \text{ ist,} \\ (b_i a, y_i, z_i + 1) & \text{falls } b_i \in \{c + 1, \dots, 2c\} \text{ ist,} \\ (b_i g, y_i + 1, z_i) & \text{falls } b_i \in \{2c + 1, \dots, q - 1\} \text{ ist.} \end{cases}$$

Aufgabe 2. Finden Sie x aus der Kongruenz

$$17 \equiv 5^x \pmod{43}$$

mit Hilfe des Pollard- ρ -Algorithmus. Benutzen Sie

- (a) mod 3-Gesetz,
- (b) Würfel-Gesetz, wobei 2, 2, 1, 1, 0, 0, 2, 0, 1, 1, 2, 0, 1, 0, 1 eine Reihe der “zufälligen” Zahlen ist,
- (c) 3-Stück-Gesetz.
- (d) Konstruieren Sie selbst eine Reihe der zufälligen Zahlen aus der Menge $\{0,1,2\}$ und lösen die Kongruenz mit dem Würfel-Gesetz.

Aufgabe 3. Finden Sie x aus der Kongruenz

$$74 \equiv 2^x \pmod{163}$$

mit Hilfe der 3 Varianten des Pollard- ρ -Algorithmus.