Implementacja algorytmu sprawdzającego pierwszość liczby w czasie wielomianowym

Martyna Siejba

Uniwersytet Wrocławski

29 kwietnia 2019

Idea

Małe Twierdzenie Fermata

Niech $a, n \in \mathbb{N}$ takie, że $NWD(a, n) = 1$. Wówczas zachodzi implikacja, że jeśli $a^{n-1} = 1 \ (mod\ n)$, to p jest liczbą pierwszą.

Idea

Małe Twierdzenie Fermata

Niech $a, n \in \mathbb{N}$ takie, że $NWD(a, n) = 1$. Wówczas zachodzi implikacja, że jeśli $a^{n-1} = 1 \,(mod\, n)$, to p jest liczbą pierwszą.

Uogólnione Twierdzenia Fermata

Niech a, $n \in \mathbb{N}$, $n \geqslant 2$ i $NWD(a, n) = 1$. Wtedy n jest pierwsza wtedy i tylko wtedy, gdy $(X + a)^n = X^n + a \,(mod\, n)$.

Idea

Małe Twierdzenie Fermata

Niech $a, n \in \mathbb{N}$ takie, że $NWD(a, n) = 1$. Wówczas zachodzi implikacja, że jeśli $a^{n-1} = 1 \,(mod\ n)$, to p jest liczbą pierwszą.

Uogólnione Twierdzenia Fermata

Niech a, $n \in \mathbb{N}$, $n \geqslant 2$ i $NWD(a, n) = 1$. Wtedy n jest pierwsza wtedy i tylko wtedy, gdy $(X + a)^n = X^n + a \,(mod\ n)$.

- Zamiast $(X + a)^n = X^n + a \,(mod\ n)$ rozpatrzmy kongruencję $(X + a)^n = X^n + a \,(mod\ X^r - 1,\ n)$, dla pewnego $r \in \mathbb{N}$.

Idea

$$(X + a)^n = X^n + a \, (mod \; X^r - 1, \; n)$$

Problem

Istnieją liczby złożone, dla których kongruencja może zachodzić.

Idea

$$(X + a)^n = X^n + a \, (mod \; X^r - 1, \; n)$$

Problem

Istnieją liczby złożone, dla których kongruencja może zachodzić.

Rozwiązanie

Jeśli sprawdzimy odpowiednio wiele różnych a jesteśmy w stanie stwierdzić, czy n jest potęgą liczby pierwszej. Niech ℓ oznacza liczbę rożnych sprawdzanych a.

Idea

$$(X + a)^n = X^n + a \, (mod \; X^r - 1, \; n)$$

Problem

Istnieją liczby złożone, dla których kongruencja może zachodzić.

Rozwiązanie

Jeśli sprawdzimy odpowiednio wiele różnych a jesteśmy w stanie stwierdzić, czy n jest potęgą liczby pierwszej. Niech ℓ oznacza liczbę rożnych sprawdzanych a.

Agrawal, Kayal i Saxena w *PRIMES is in P* znajdują odpowiednie r i ℓ.

Algorytm

1: **if** istnieje takie $a \in \mathbb{N}, b > 1$, że $a^b = n$ **then** ▷ Krok 1.
2: **return** ZŁOŻONA
3: **end if**

Algorytm

1: **if** istnieje takie $a \in \mathbb{N}, b > 1$, że $a^b = n$ **then** ▷ Krok 1.
2: **return** ZŁOŻONA
3: **end if**
4: $r \leftarrow$ najmniejsza liczba taka, że zachodzi $o_r(n) > log^2 n$ ▷ Krok 2.

Algorytm

1: **if** istnieje takie $a \in \mathbb{N}, b > 1$, że $a^b = n$ **then** ▷ Krok 1.
2: **return** ZŁOŻONA
3: **end if**
4: $r \leftarrow$ najmniejsza liczba taka, że zachodzi $o_r(n) > log^2 n$ ▷ Krok 2.
5: **if** istnieje $a \leqslant r$ takie, że $1 < NWD(a, n) < n$ **then** ▷ Krok 3.
6: **return** ZŁOŻONA
7: **end if**

Algorytm

1: **if** istnieje takie $a \in \mathbb{N}, b > 1$, że $a^b = n$ **then** ▷ Krok 1.
2: **return** ZŁOŻONA
3: **end if**
4: $r \leftarrow$ najmniejsza liczba taka, że zachodzi $o_r(n) > log^2 n$ ▷ Krok 2.
5: **if** istnieje $a \leqslant r$ takie, że $1 < NWD(a, n) < n$ **then** ▷ Krok 3.
6: **return** ZŁOŻONA
7: **end if**
8: **if** $n \leqslant r$ **then** ▷ Krok 4.
9: **return** PIERWSZA
10: **end if**

Algorytm

1: **if** istnieje takie $a \in \mathbb{N}, b > 1$, że $a^b = n$ **then** ▷ Krok 1.
2: **return** ZŁOŻONA
3: **end if**
4: $r \leftarrow$ najmniejsza liczba taka, że zachodzi $o_r(n) > log^2 n$ ▷ Krok 2.
5: **if** istnieje $a \leqslant r$ takie, że $1 < NWD(a, n) < n$ **then** ▷ Krok 3.
6: **return** ZŁOŻONA
7: **end if**
8: **if** $n \leqslant r$ **then** ▷ Krok 4.
9: **return** PIERWSZA
10: **end if**
11: **for** $a \leftarrow 1$ **to** $\lfloor \sqrt{\phi(r)}\, log\, n \rfloor$ **do** ▷ Krok 5.
12: **if** $(X + a)^n \neq X^n + a\, (mod\, X^r - 1,\, n)$ **then** ▷ Krok 6.
13: **return** ZŁOŻONA
14: **end if**
15: **end for**
16: **return** PIERWSZA ▷ Krok 7.

Złożoność obliczeniowa

Lemat

Zachodzi nierówność $r \leqslant \max\{3, \lceil \log^5 n \rceil\}$.

- krok 1: $O(log\ n \cdot (log\ n \cdot (log\ b \cdot log\ n))) = O(\mathbf{\log^2 n \cdot \log\log n})$.
- krok 2: $O(log^5 n \cdot (log^2 n \cdot log\ r)) = O(\mathbf{\log^7 n \cdot \log\log n})$.
- krok 3: $O(r \cdot (log\ n + log^2 r)) = O(\mathbf{\log^6 n} + log\ n \cdot log^2 log\ n))$.
- krok 4: $O(log\ n)$.
- krok 5:
  - krok 6: $O(logn \cdot (r \cdot log\ r \cdot log\ n)) = O(\mathbf{\log^7 n \cdot \log\log n})$.

$O(\sqrt{\phi(r)}\ log\ n \cdot log^7 n \cdot log\ log\ n) \subseteq O(\sqrt{r}\ log\ n \cdot log^7 n \cdot log\ log\ n) \subseteq O(log^{\frac{5}{2}} n \cdot log^8 n) \subseteq O(\mathbf{\log^{\frac{21}{2}} n \cdot \log\log n})$.

Otrzymujemy ostateczną złożoność $O(log^{\frac{21}{2}} n \cdot log\ log\ n)$.

Dowód poprawności

Założenia o zmiennych r i n w kroku 5.:

- n nie jest potęgą liczby pierwszej;
- $\forall_{r'<r}\ o_r(n) \leqslant log^2 n$;
- $n > r$;
- $\forall_{a \leqslant r}\ NWD(a, n) = 1$.

Dowód poprawności

Założenia o zmiennych r i n w kroku 5.:

- n nie jest potęgą liczby pierwszej;
- $\forall_{r'<r}\ o_r(n) \leqslant log^2 n$;
- $n > r$;
- $\forall_{a \leqslant r}\ NWD(a, n) = 1$.

Idea dowodu

W dowodzie do powyższych założeń dodamy założenie, że istnieje pierwsza liczba $p < n$ taka, że $p \mid n$. Na podstawie jej istnienia zdefiniujemy zbiory, których własności doprowadzą do sprzeczności.

Dowód poprawności

Niech

- $I = \{(\frac{n}{p})^i \cdot p^j \mid i, j \geqslant 0\}$,
- $P = \{\prod_{a=0}^{\ell} (X + a)^{e_a} \mid e_a \geqslant 0\}$,

Dowód poprawności

Niech

- $I = \{(\frac{n}{p})^i \cdot p^j \mid i, j \geqslant 0\}$,
- $P = \{\prod_{a=0}^{\ell} (X + a)^{e_a} \mid e_a \geqslant 0\}$,
- $G = \{i \,(mod\, r) \mid i \in I\}$, $t = |G|$,

Dowód poprawności

Niech

- $I = \{(\frac{n}{p})^i \cdot p^j \mid i, j \geqslant 0\}$,
- $P = \{\prod_{a=0}^{\ell} (X + a)^{e_a} \mid e_a \geqslant 0\}$,
- $G = \{i \,(mod\, r) \mid i \in I\}$, $t = |G|$,
- $h(X) \in \mathbb{Z}_p[X]$ oraz $h(X) \mid Q_r(X)$ i $deg(h) = o_r(p)$,

Dowód poprawności

Niech

- $I = \{(\frac{n}{p})^i \cdot p^j \mid i, j \geqslant 0\}$,
- $P = \{\prod_{a=0}^{\ell} (X + a)^{e_a} \mid e_a \geqslant 0\}$,
- $G = \{i \,(mod\, r) \mid i \in I\}$, $t = |G|$,
- $h(X) \in \mathbb{Z}_p[X]$ oraz $h(X) \mid Q_r(X)$ i $deg(h) = o_r(p)$,
- $\mathcal{G} = \{q(X) \,(mod\, h(X),\, p) \,|q(X) \in P\}$

Dowód poprawności

Szacowanie dolne mocy $\mathcal{G}$

Pokazujemy, że $|\mathcal{G}| \geqslant \binom{t+\ell}{t-1}$.
Dowód przebiega poprzez pokazanie, że dwa różne wielomiany stopnia mniejszego niż t z P odpowiadają różnym wielomianom w $\mathcal{G}$, a następnie oszacowanie liczby tych wielomianów przez $\binom{t+\ell}{t-1}$.

Szacowanie górne mocy $\mathcal{G}$

Dowodzimy, że $|\mathcal{G}| \leqslant n^{\sqrt{t}}$.
W dowodzie, wykorzystując własności elementów zbioru I, znajdujemy wielomian, dla którego wszystkie elementy $\mathcal{G}$ są pierwiastkami i ograniczamy z góry jego stopień.

Sprzeczność otrzymujemy pokazując, że $\binom{t+\ell}{t-1} > n^{\sqrt{t}}$.

Implementacja

- Algorytm zaimplementowałam w języku C++ (standard C++11).
- Wykorzystana biblioteka NTL w wersji 11.3.2.