1. Teoria liczb

Grzegorz Kosiorowski

Uniwersytet Ekonomiczny w Krakowie

Teoria liczb - wstęp

Teoria liczb to „matematyka matematyki” zajmująca się badaniem własności liczb - przede wszystkim całkowitych (w tym rozdziale, jeśli nie będzie wyraźnie powiedziane inaczej, wszystkie liczby są całkowite - i w czasie rozwiązywania zadań tylko takie mogą występować).

Teoria liczb - wstęp

Teoria liczb to „matematyka matematyki” zajmująca się badaniem własności liczb - przede wszystkim całkowitych (w tym rozdziale, jeśli nie będzie wyraźnie powiedziane inaczej, wszystkie liczby są całkowite - i w czasie rozwiązywania zadań tylko takie mogą występować). Kiedyś wydawało się, że to dziedzina najbardziej abstrakcyjna z możliwych i praktycznych zastosowań mieć nie będzie.

Teoria liczb - wstęp

Teoria liczb to „matematyka matematyki” zajmująca się badaniem własności liczb - przede wszystkim całkowitych (w tym rozdziale, jeśli nie będzie wyraźnie powiedziane inaczej, wszystkie liczby są całkowite - i w czasie rozwiązywania zadań tylko takie mogą występować). Kiedyś wydawało się, że to dziedzina najbardziej abstrakcyjna z możliwych i praktycznych zastosowań mieć nie będzie. Dziś już wiemy, że to nieprawda.

Teoria liczb - zastosowania

Teoria liczb - zastosowania

- Współczesne systemy szyfrowania informacji (opierają się na trudności faktoryzacji liczb całkowitych - dokładniej omówimy to w kolejnym rozdziale).

Teoria liczb - zastosowania

- Współczesne systemy szyfrowania informacji (opierają się na trudności faktoryzacji liczb całkowitych - dokładniej omówimy to w kolejnym rozdziale).
- Efektywne metody kompresji danych.

Teoria liczb - zastosowania

- Współczesne systemy szyfrowania informacji (opierają się na trudności faktoryzacji liczb całkowitych - dokładniej omówimy to w kolejnym rozdziale).
- Efektywne metody kompresji danych.
- Generowanie liczb pseudolosowych.

Teoria liczb - zastosowania

- Współczesne systemy szyfrowania informacji (opierają się na trudności faktoryzacji liczb całkowitych - dokładniej omówimy to w kolejnym rozdziale).
- Efektywne metody kompresji danych.
- Generowanie liczb pseudolosowych.
- Kodowanie informacji z autowykrywaniem błędów (np. numery kont bankowych i kart kredytowych, cyfrowy zapis muzyki).

Teoria liczb - zastosowania

- Współczesne systemy szyfrowania informacji (opierają się na trudności faktoryzacji liczb całkowitych - dokładniej omówimy to w kolejnym rozdziale).
- Efektywne metody kompresji danych.
- Generowanie liczb pseudolosowych.
- Kodowanie informacji z autowykrywaniem błędów (np. numery kont bankowych i kart kredytowych, cyfrowy zapis muzyki).
- Automatyczne generowanie (dającej się słuchać) muzyki (Brian Eno).

Teoria liczb - zastosowania

- Współczesne systemy szyfrowania informacji (opierają się na trudności faktoryzacji liczb całkowitych - dokładniej omówimy to w kolejnym rozdziale).
- Efektywne metody kompresji danych.
- Generowanie liczb pseudolosowych.
- Kodowanie informacji z autowykrywaniem błędów (np. numery kont bankowych i kart kredytowych, cyfrowy zapis muzyki).
- Automatyczne generowanie (dającej się słuchać) muzyki (Brian Eno).
- Zastosowania teorii liczb we współczesnej fizyce teoretycznej: http://empslocal.ex.ac.uk/people/staff/mrwatkin/zeta/physics.htm

Dzielenie w zbiorze liczb całkowitych

Dodawanie, odejmowanie i mnożenie w ramach liczb całkowitych nie ma szczególnie interesujących własności. Jednak ciekawe problemy pojawiają się przy dzieleniu, gdyż wynik dzielenia dwu liczb całkowitych nie musi być całkowity.

Iloraz i reszta z dzielenia

Jeśli dla pewnych liczb całkowitych a, b istnieją liczby q, r takie, że $a = bq + r$ i $0 \leqslant r < b$, to q nazywamy *ilorazem* liczb a i b, a r - *resztą* z dzielenia a przez b. Zapisujemy $r = a \mod b$.

Dzielenie w zbiorze liczb całkowitych

Dodawanie, odejmowanie i mnożenie w ramach liczb całkowitych nie ma szczególnie interesujących własności. Jednak ciekawe problemy pojawiają się przy dzieleniu, gdyż wynik dzielenia dwu liczb całkowitych nie musi być całkowity.

Iloraz i reszta z dzielenia

Jeśli dla pewnych liczb całkowitych a, b istnieją liczby q, r takie, że $a = bq + r$ i $0 \leqslant r < b$, to q nazywamy *ilorazem* liczb a i b, a r - *resztą* z dzielenia a przez b. Zapisujemy $r = a \mod b$.

$$17 = 5 \cdot 3 + 2$$

$$2 = 17 \mod 5$$

Podzielność

Podzielność

Mówimy, że b *dzieli* a (lub a *jest podzielne* przez b, b jest *dzielnikiem* a, a jest *wielokrotnością* b), jeśli istnieje q takie, że $a = bq$, czyli, gdy $a \mod b = 0$. Zapisujemy $b|a$.

Podzielność

Podzielność

Mówimy, że b *dzieli* a (lub a *jest podzielne* przez b, b jest *dzielnikiem* a, a jest *wielokrotnością* b), jeśli istnieje q takie, że $a = bq$, czyli, gdy $a \mod b = 0$. Zapisujemy $b|a$.

Przykład: $3|6$.

Twierdzenia o podzielności

Twierdzenia o podzielności

Dla dowolnych liczb $a, b, c \neq 0$ zachodzi:
a) jeśli $a|b$ to $a|bc$,
b) jeśli $a|b$ i $b|c$ to $a|c$,
c) jeśli $a|b$ i $a|c$ to $a|(b+c)$.

Największy wspólny dzielnik

Największy wspólny dzielnik

Największy wspólny dzielnik niezerowych liczb a i b (zapisywany jako NWD(a, b)) to największa liczba d taka, że $d|a$ i $d|b$.

Największy wspólny dzielnik

Największy wspólny dzielnik

Największy wspólny dzielnik niezerowych liczb a i b (zapisywany jako NWD(a, b)) to największa liczba d taka, że $d|a$ i $d|b$.

Twierdzonko

Dla niezerowych liczb a, b zachodzi:
$NWD(a, 1) =$

Największy wspólny dzielnik

Największy wspólny dzielnik

Największy wspólny dzielnik niezerowych liczb a i b (zapisywany jako NWD(a, b)) to największa liczba d taka, że $d|a$ i $d|b$.

Twierdzonko

Dla niezerowych liczb a, b zachodzi:
$NWD(a, 1) = 1$, $NWD(a, a) =$

Największy wspólny dzielnik

Największy wspólny dzielnik

Największy wspólny dzielnik niezerowych liczb a i b (zapisywany jako NWD(a, b)) to największa liczba d taka, że $d|a$ i $d|b$.

Twierdzonko

Dla niezerowych liczb a, b zachodzi:
$NWD(a, 1) = 1$, $NWD(a, a) = a$, $NWD(a, b) = NWD(b, a)$.

Największy wspólny dzielnik

Największy wspólny dzielnik

Największy wspólny dzielnik niezerowych liczb a i b (zapisywany jako NWD(a, b)) to największa liczba d taka, że $d|a$ i $d|b$.

Twierdzonko

Dla niezerowych liczb a, b zachodzi:
$NWD(a, 1) = 1$, $NWD(a, a) = a$, $NWD(a, b) = NWD(b, a)$.

Przykłady: $NWD(9, 6) =$

Największy wspólny dzielnik

Największy wspólny dzielnik

Największy wspólny dzielnik niezerowych liczb a i b (zapisywany jako NWD(a, b)) to największa liczba d taka, że $d|a$ i $d|b$.

Twierdzonko

Dla niezerowych liczb a, b zachodzi:
$NWD(a, 1) = 1$, $NWD(a, a) = a$, $NWD(a, b) = NWD(b, a)$.

Przykłady: $NWD(9, 6) = 3$, $NWD(36, 12) =$

Największy wspólny dzielnik

Największy wspólny dzielnik

Największy wspólny dzielnik niezerowych liczb a i b (zapisywany jako NWD(a, b)) to największa liczba d taka, że $d|a$ i $d|b$.

Twierdzonko

Dla niezerowych liczb a, b zachodzi:
$NWD(a, 1) = 1$, $NWD(a, a) = a$, $NWD(a, b) = NWD(b, a)$.

Przykłady: $NWD(9, 6) = 3$, $NWD(36, 12) = 12$, $NWD(33, 26) =$

Największy wspólny dzielnik

Największy wspólny dzielnik

Największy wspólny dzielnik niezerowych liczb a i b (zapisywany jako NWD(a, b)) to największa liczba d taka, że $d|a$ i $d|b$.

Twierdzonko

Dla niezerowych liczb a, b zachodzi:
$NWD(a, 1) = 1$, $NWD(a, a) = a$, $NWD(a, b) = NWD(b, a)$.

Przykłady: $NWD(9, 6) = 3$, $NWD(36, 12) = 12$, $NWD(33, 26) = 1$,

Największy wspólny dzielnik

Największy wspólny dzielnik

Największy wspólny dzielnik niezerowych liczb a i b (zapisywany jako NWD(a, b)) to największa liczba d taka, że $d|a$ i $d|b$.

Twierdzonko

Dla niezerowych liczb a, b zachodzi:
$NWD(a, 1) = 1$, $NWD(a, a) = a$, $NWD(a, b) = NWD(b, a)$.

Przykłady: $NWD(9, 6) = 3$, $NWD(36, 12) = 12$, $NWD(33, 26) = 1$,
Jednak dla większych liczb znajdowanie największego wspólnego dzielnika wymaga systematycznego podejścia.

Algorytm Euklidesa

Poniżej algorytm, który ma już około 2300 lat, pochodzący z „Elementów” Euklidesa. Służy do znajdowania największego wspólnego dzielnika liczb całkowitych.

Algorytm Euklidesa

Dane: Liczby całkowite dodatnie a, b.
Zmienne: r - liczba całkowita.

Algorytm Euklidesa

Poniżej algorytm, który ma już około 2300 lat, pochodzący z „Elementów" Euklidesa. Służy do znajdowania największego wspólnego dzielnika liczb całkowitych.

Algorytm Euklidesa

Dane: Liczby całkowite dodatnie a, b.
Zmienne: r - liczba całkowita.

- I. Dopóki $b \neq 0$ wykonuj:

Algorytm Euklidesa

Poniżej algorytm, który ma już około 2300 lat, pochodzący z „Elementów" Euklidesa. Służy do znajdowania największego wspólnego dzielnika liczb całkowitych.

Algorytm Euklidesa

Dane: Liczby całkowite dodatnie a, b.
Zmienne: r - liczba całkowita.

- I. Dopóki $b \neq 0$ wykonuj:
- Ia. $r := a \mod b$.

Algorytm Euklidesa

Poniżej algorytm, który ma już około 2300 lat, pochodzący z „Elementów” Euklidesa. Służy do znajdowania największego wspólnego dzielnika liczb całkowitych.

Algorytm Euklidesa

Dane: Liczby całkowite dodatnie a, b.
Zmienne: r - liczba całkowita.

- I. Dopóki $b \neq 0$ wykonuj:
- Ia. $r := a \mod b$.
- Ib. $a := b$, $b := r$.

Algorytm Euklidesa

Poniżej algorytm, który ma już około 2300 lat, pochodzący z „Elementów" Euklidesa. Służy do znajdowania największego wspólnego dzielnika liczb całkowitych.

Algorytm Euklidesa

Dane: Liczby całkowite dodatnie a, b.
Zmienne: r - liczba całkowita.

- I. Dopóki $b \neq 0$ wykonuj:
- Ia. $r := a \mod b$.
- Ib. $a := b$, $b := r$.
- **Rezultat**: Na końcu działania algorytmu a jest największym wspólnym dzielnikiem danych na początku liczb.

Algorytm Euklidesa - przykład

Przykład

Znaleźć $NWD(888, 1104)$.

Algorytm Euklidesa - przykład

Przykład

Znaleźć $NWD(888, 1104)$.

Rozpoczynamy od podstawienia $a = 888$, $b = 1104$.

Algorytm Euklidesa - przykład

Przykład

Znaleźć $NWD(888, 1104)$.

Rozpoczynamy od podstawienia $a = 888$, $b = 1104$.

Wyniki algorytmu będziemy zapisywać w poniższej tabeli:

Nr kroku	r	a	b
1			
2			
...			

Algorytm Euklidesa - przykład

Przykład

Znaleźć $NWD(888, 1104)$.

Algorytm Euklidesa - przykład

Przykład

Znaleźć $NWD(888, 1104)$.

Skoro $b \neq 0$, to w pierwszym kroku $r = a \mod b = 888$ (bo $a = 0 \cdot 1104 + 888$)

Algorytm Euklidesa - przykład

Przykład

Znaleźć $NWD(888, 1104)$.

Skoro $b \neq 0$, to w pierwszym kroku $r = a \mod b = 888$ (bo $a = 0 \cdot 1104 + 888$)
Następnie $a := b = 1104$, a $b := r = 888$.

Algorytm Euklidesa - przykład

Przykład

Znaleźć $NWD(888, 1104)$.

Skoro $b \neq 0$, to w pierwszym kroku $r = a \mod b = 888$ (bo $a = 0 \cdot 1104 + 888$)
Następnie $a := b = 1104$, a $b := r = 888$. Zatem pierwszy krok jedynie zamienił nam kolejnością a i b (dlatego warto zaczynać algorytm Euklidesa tak, by $a > b$)

Algorytm Euklidesa - przykład

Przykład

Znaleźć $NWD(888, 1104)$.

Skoro $b \neq 0$, to w pierwszym kroku $r = a \mod b = 888$ (bo $a = 0 \cdot 1104 + 888$)
Następnie $a := b = 1104$, a $b := r = 888$. Zatem pierwszy krok jedynie zamienił nam kolejnością a i b (dlatego warto zaczynać algorytm Euklidesa tak, by $a > b$)

Nr kroku	r	a	b
1	888	1104	888
2			

Algorytm Euklidesa - przykład

Przykład

Znaleźć $NWD(888, 1104)$.

Algorytm Euklidesa - przykład

Przykład

Znaleźć $NWD(888, 1104)$.

Nadal $b \neq 0$, więc $r = 1104 \mod 888 = 216$ (bo $1104 = 1 \cdot 888 + 216$)

Algorytm Euklidesa - przykład

Przykład

Znaleźć $NWD(888, 1104)$.

Nadal $b \neq 0$, więc $r = 1104 \mod 888 = 216$ (bo $1104 = 1 \cdot 888 + 216$)
Następnie $a := b = 888$, a $b := r = 216$.

Algorytm Euklidesa - przykład

Przykład

Znaleźć $NWD(888, 1104)$.

Nadal $b \neq 0$, więc $r = 1104 \mod 888 = 216$ (bo $1104 = 1 \cdot 888 + 216$)

Następnie $a := b = 888$, a $b := r = 216$.

Nr kroku	r	a	b
1	888	1104	888
2	216	888	216

Algorytm Euklidesa - przykład

Przykład

Znaleźć $NWD(888, 1104)$.

Algorytm Euklidesa - przykład

Przykład

Znaleźć $NWD(888, 1104)$.

Nadal $b \neq 0$, więc $r = 888 \mod 216 = 24$ (bo $888 = 4 \cdot 216 + 24$)

Algorytm Euklidesa - przykład

Przykład

Znaleźć $NWD(888, 1104)$.

Nadal $b \neq 0$, więc $r = 888 \mod 216 = 24$ (bo $888 = 4 \cdot 216 + 24$)
Następnie $a := b = 216$, a $b := r = 24$.

Algorytm Euklidesa - przykład

Przykład

Znaleźć $NWD(888, 1104)$.

Nadal $b \neq 0$, więc $r = 888 \mod 216 = 24$ (bo $888 = 4 \cdot 216 + 24$)
Następnie $a := b = 216$, a $b := r = 24$.

Nr kroku	r	a	b
1	888	1104	888
2	216	888	216
3	24	216	24

Algorytm Euklidesa - przykład

Przykład

Znaleźć $NWD(888, 1104)$.

Algorytm Euklidesa - przykład

Przykład

Znaleźć $NWD(888, 1104)$.

Nadal $b \neq 0$, więc $r = 216 \mod 24 = 0$ (bo $216 = 9 \cdot 24$)

Algorytm Euklidesa - przykład

Przykład

Znaleźć $NWD(888, 1104)$.

Nadal $b \neq 0$, więc $r = 216 \mod 24 = 0$ (bo $216 = 9 \cdot 24$)
Następnie $a := b = 24$, a $b := r = 0$.

Algorytm Euklidesa - przykład

Przykład

Znaleźć $NWD(888, 1104)$.

Nadal $b \neq 0$, więc $r = 216 \mod 24 = 0$ (bo $216 = 9 \cdot 24$)
Następnie $a := b = 24$, a $b := r = 0$. $b = 0$, więc algorytm jest zakończony i $NWD(888, 1104) = a = 24$.

Nr kroku	r	a	b
1	888	1104	888
2	216	888	216
3	24	216	24
4	0	24	0

Rozszerzony algorytm Euklidesa

W kryptografii przydaje się tzw. rozszerzony algorytm Euklidesa, który znajduje liczby x i y z poniższego twierdzenia.

Rozszerzony algorytm Euklidesa

W kryptografii przydaje się tzw. rozszerzony algorytm Euklidesa, który znajduje liczby x i y z poniższego twierdzenia.

Twierdzenie o kombinacji liniowej liczb naturalnych

Dla dowolnych liczb całkowitych dodatnich a i b istnieją liczby całkowite x i y takie, że $ax + by = NWD(a, b)$.

Rozszerzony algorytm Euklidesa

Rozszerzony algorytm Euklidesa - część I

Dane: Liczby całkowite dodatnie a, b.

Zmienne: r_i, q_i, x_i,y_i - ciągi liczb całkowitych, i-licznik pętli.

Rozszerzony algorytm Euklidesa

Rozszerzony algorytm Euklidesa - część I

Dane: Liczby całkowite dodatnie a, b.

Zmienne: r_i, q_i, x_i,y_i - ciągi liczb całkowitych, i-licznik pętli.

- I. $r_0 := a$, $r_1 := b$,$i := 1$.

Rozszerzony algorytm Euklidesa

Rozszerzony algorytm Euklidesa - część I

Dane: Liczby całkowite dodatnie a, b.

Zmienne: r_i, q_i, x_i,y_i - ciągi liczb całkowitych, i-licznik pętli.

- I. $r_0 := a$, $r_1 := b$,$i := 1$.
- II. Dopóki $r_i \neq 0$ wykonuj:

Rozszerzony algorytm Euklidesa

Rozszerzony algorytm Euklidesa - część I

Dane: Liczby całkowite dodatnie a, b.

Zmienne: r_i, q_i, x_i,y_i - ciągi liczb całkowitych, i-licznik pętli.

- I. $r_0 := a$, $r_1 := b$,$i := 1$.
- II. Dopóki $r_i \neq 0$ wykonuj:
- IIa. $i := i + 1$.

Rozszerzony algorytm Euklidesa

Rozszerzony algorytm Euklidesa - część I

Dane: Liczby całkowite dodatnie a, b.

Zmienne: r_i, q_i, x_i,y_i - ciągi liczb całkowitych, i-licznik pętli.

- I. $r_0 := a$, $r_1 := b$,$i := 1$.
- II. Dopóki $r_i \neq 0$ wykonuj:
- IIa. $i := i + 1$.
- IIb. $r_i := r_{i-2} \mod r_{i-1}$, $q_{i-1} := (r_{i-2} - r_i)/r_{i-1}$.

Rozszerzony algorytm Euklidesa

Rozszerzony algorytm Euklidesa - część I

Dane: Liczby całkowite dodatnie a, b.

Zmienne: r_i, q_i, x_i,y_i - ciągi liczb całkowitych, i-licznik pętli.

- I. $r_0 := a$, $r_1 := b$,$i := 1$.
- II. Dopóki $r_i \neq 0$ wykonuj:
- IIa. $i := i + 1$.
- IIb. $r_i := r_{i-2} \mod r_{i-1}$, $q_{i-1} := (r_{i-2} - r_i)/r_{i-1}$.
- Zauważmy, że w momencie zakończenia działania tej pętli obliczyliśmy już r_i =NWD(a, b) i dla wszystkich $k < i - 1$ zachodzi $r_k - q_{k+1} r_{k+1} = r_{k+2}$.

Rozszerzony algorytm Euklidesa - część II

Rozszerzony algorytm Euklidesa

Rozszerzony algorytm Euklidesa - część II

- III. $i := i - 1$, $x_i := 0$, $y_i := 1$.

Rozszerzony algorytm Euklidesa

Rozszerzony algorytm Euklidesa - część II

- III. $i := i - 1$, $x_i := 0$, $y_i := 1$.
- IV. Dopóki $i > 1$ wykonuj:

Rozszerzony algorytm Euklidesa

Rozszerzony algorytm Euklidesa - część II

- III. $i := i - 1$, $x_i := 0$, $y_i := 1$.
- IV. Dopóki $i > 1$ wykonuj:
- IVa. $i := i - 1$.

Rozszerzony algorytm Euklidesa

Rozszerzony algorytm Euklidesa - część II

- III. $i := i - 1$, $x_i := 0$, $y_i := 1$.
- IV. Dopóki $i > 1$ wykonuj:
- IVa. $i := i - 1$.
- IVb. $x_i := y_{i+1}$, $y_i := x_{i+1} - q_i x_i$.

Rozszerzony algorytm Euklidesa

Rozszerzony algorytm Euklidesa - część II

- III. $i := i - 1$, $x_i := 0$, $y_i := 1$.
- IV. Dopóki $i > 1$ wykonuj:
- IVa. $i := i - 1$.
- IVb. $x_i := y_{i+1}$, $y_i := x_{i+1} - q_i x_i$.
- **Rezultat**: (x_1, y_1) są odpowiednią parą z twierdzenia. NWD jest obliczone wcześniej.

Rozszerzony algorytm Euklidesa - przykład

Przykład

Znaleźć NWD(234, 123) oraz liczby całkowite x, y takie, że $234x + 123y = NWD(234, 123)$.

Rozszerzony algorytm Euklidesa - przykład

Przykład

Znaleźć NWD$(234, 123)$ oraz liczby całkowite x, y takie, że $234x + 123y = NWD(234, 123)$.

Rozpoczynamy od podstawienia $a = 234$, $b = 123$.

Rozszerzony algorytm Euklidesa - przykład

Przykład

Znaleźć NWD(234, 123) oraz liczby całkowite x, y takie, że $234x + 123y = NWD(234, 123)$.

Rozpoczynamy od podstawienia $a = 234$, $b = 123$.
Wyniki algorytmu będziemy zapisywać w poniższej tabeli:

i	r_i	q_i	x_i	y_i
0	?			
1	?	?	?	?
2	?	?	?	?
...	...	...	...	...

Rozszerzony algorytm Euklidesa - przykład

Przykład

Znaleźć NWD$(234, 123)$ oraz liczby całkowite x, y takie, że $234x + 123y = NWD(234, 123)$.

Rozszerzony algorytm Euklidesa - przykład

Przykład

Znaleźć NWD(234, 123) oraz liczby całkowite x, y takie, że $234x + 123y = NWD(234, 123)$.

I etap algorytmu to wpisanie do tabeli w odpowiednie miejsca liczb a i b (i ustawienie licznika $i = 1$):

i	r_i	q_i	x_i	y_i
0	234			
1	123	?	?	?
2	?	?	?	?
...	...	...	...	...

Rozszerzony algorytm Euklidesa - przykład

i	r_i	q_i	x_i	y_i
0	234			
1	123	?	?	?
2	?	?	?	?
...	...	...	...	...

$r_i = r_1 \neq 0$, więc kontynuujemy etap II:

Rozszerzony algorytm Euklidesa - przykład

i	r_i	q_i	x_i	y_i
0	234			
1	123	?	?	?
2	?	?	?	?
...	...	...	...	...

$r_i = r_1 \neq 0$, więc kontynuujemy etap II: $i = 2$,

Rozszerzony algorytm Euklidesa - przykład

i	r_i	q_i	x_i	y_i
0	234			
1	123	?	?	?
2	?	?	?	?
...	...	...	...	...

$r_i = r_1 \neq 0$, więc kontynuujemy etap II: $i = 2$, $234 = 1 \cdot 123 + 111$,

Rozszerzony algorytm Euklidesa - przykład

i	r_i	q_i	x_i	y_i
0	234			
1	123	?	?	?
2	?	?	?	?
...	...	...	...	...

$r_i = r_1 \neq 0$, więc kontynuujemy etap II: $i = 2$, $234 = 1 \cdot 123 + 111$, stąd $r_2 = 111$

Rozszerzony algorytm Euklidesa - przykład

i	r_i	q_i	x_i	y_i
0	234			
1	123	?	?	?
2	?	?	?	?
...	...	...	...	...

$r_i = r_1 \neq 0$, więc kontynuujemy etap II: $i = 2$, $234 = 1 \cdot 123 + 111$, stąd $r_2 = 111$ i $q_1 = 1$.

Rozszerzony algorytm Euklidesa - przykład

$r_i = r_1 \neq 0$, więc kontynuujemy etap II: $i = 2$, $234 = 1 \cdot 123 + 111$, stąd $r_2 = 111$ i $q_1 = 1$.

i	r_i	q_i	x_i	y_i
0	234			
1	123	1	?	?
2	111	?	?	?
3	?	?	?	?
...	...	...	...	...

Rozszerzony algorytm Euklidesa - przykład

i	r_i	q_i	x_i	y_i
0	234			
1	123	1	?	?
2	111	?	?	?
3	?	?	?	?
...	...	...	...	...

$r_i = r_2 \neq 0$, więc kontynuujemy etap II:

Rozszerzony algorytm Euklidesa - przykład

i	r_i	q_i	x_i	y_i
0	234			
1	123	1	?	?
2	111	?	?	?
3	?	?	?	?
...	...	...	...	...

$r_i = r_2 \neq 0$, więc kontynuujemy etap II: $i = 3$,

Rozszerzony algorytm Euklidesa - przykład

i	r_i	q_i	x_i	y_i
0	234			
1	123	1	?	?
2	111	?	?	?
3	?	?	?	?
...	...	...	...	...

$r_i = r_2 \neq 0$, więc kontynuujemy etap II: $i = 3$, $123 = 1 \cdot 111 + 12$,

Rozszerzony algorytm Euklidesa - przykład

i	r_i	q_i	x_i	y_i
0	234			
1	123	1	?	?
2	111	?	?	?
3	?	?	?	?
...	...	...	...	...

$r_i = r_2 \neq 0$, więc kontynuujemy etap II: $i = 3$, $123 = 1 \cdot 111 + 12$, stąd $r_3 = 12$

Rozszerzony algorytm Euklidesa - przykład

i	r_i	q_i	x_i	y_i
0	234			
1	123	1	?	?
2	111	?	?	?
3	?	?	?	?
...	...	...	...	...

$r_i = r_2 \neq 0$, więc kontynuujemy etap II: $i = 3$, $123 = 1 \cdot 111 + 12$, stąd $r_3 = 12$ i $q_2 = 1$.

Rozszerzony algorytm Euklidesa - przykład

$r_i = r_2 \neq 0$, więc kontynuujemy etap II: $i = 3$, $123 = 1 \cdot 111 + 12$, stąd $r_3 = 12$ i $q_2 = 1$.

i	r_i	q_i	x_i	y_i
0	234			
1	123	1	?	?
2	111	1	?	?
3	12	?	?	?
4	?	?	?	?

Rozszerzony algorytm Euklidesa - przykład

i	r_i	q_i	x_i	y_i
0	234			
1	123	1	?	?
2	111	1	?	?
3	12	?	?	?
4	?	?	?	?

$r_i = r_3 \neq 0$, więc kontynuujemy etap II:

Rozszerzony algorytm Euklidesa - przykład

i	r_i	q_i	x_i	y_i
0	234			
1	123	1	?	?
2	111	1	?	?
3	12	?	?	?
4	?	?	?	?

$r_i = r_3 \neq 0$, więc kontynuujemy etap II: $i = 4$,

Rozszerzony algorytm Euklidesa - przykład

i	r_i	q_i	x_i	y_i
0	234			
1	123	1	?	?
2	111	1	?	?
3	12	?	?	?
4	?	?	?	?

$r_i = r_3 \neq 0$, więc kontynuujemy etap II: $i = 4$, $111 = 9 \cdot 12 + 3$,

Rozszerzony algorytm Euklidesa - przykład

i	r_i	q_i	x_i	y_i
0	234			
1	123	1	?	?
2	111	1	?	?
3	12	?	?	?
4	?	?	?	?

$r_i = r_3 \neq 0$, więc kontynuujemy etap II: $i = 4$, $111 = 9 \cdot 12 + 3$, stąd $r_4 = 3$

Rozszerzony algorytm Euklidesa - przykład

i	r_i	q_i	x_i	y_i
0	234			
1	123	1	?	?
2	111	1	?	?
3	12	?	?	?
4	?	?	?	?

$r_i = r_3 \neq 0$, więc kontynuujemy etap II: $i = 4$, $111 = 9 \cdot 12 + 3$, stąd $r_4 = 3$ i $q_3 = 9$.

Rozszerzony algorytm Euklidesa - przykład

$r_i = r_3 \neq 0$, więc kontynuujemy etap II: $i = 4$, $111 = 9 \cdot 12 + 3$, stąd $r_4 = 3$ i $q_3 = 9$.

i	r_i	q_i	x_i	y_i
0	234			
1	123	1	?	?
2	111	1	?	?
3	12	9	?	?
4	3	?	?	?

Rozszerzony algorytm Euklidesa - przykład

i	r_i	q_i	x_i	y_i
0	234			
1	123	1	?	?
2	111	1	?	?
3	12	9	?	?
4	3	?	?	?

$r_i = r_4 \neq 0$, więc kontynuujemy etap II:

Rozszerzony algorytm Euklidesa - przykład

i	r_i	q_i	x_i	y_i
0	234			
1	123	1	?	?
2	111	1	?	?
3	12	9	?	?
4	3	?	?	?

$r_i = r_4 \neq 0$, więc kontynuujemy etap II: $i = 5$,

Rozszerzony algorytm Euklidesa - przykład

i	r_i	q_i	x_i	y_i
0	234			
1	123	1	?	?
2	111	1	?	?
3	12	9	?	?
4	3	?	?	?

$r_i = r_4 \neq 0$, więc kontynuujemy etap II: $i = 5$, $12 = 4 \cdot 3 + 0$,

Rozszerzony algorytm Euklidesa - przykład

i	r_i	q_i	x_i	y_i
0	234			
1	123	1	?	?
2	111	1	?	?
3	12	9	?	?
4	3	?	?	?

$r_i = r_4 \neq 0$, więc kontynuujemy etap II: $i = 5$, $12 = 4 \cdot 3 + 0$, stąd $r_5 = 0$

Rozszerzony algorytm Euklidesa - przykład

i	r_i	q_i	x_i	y_i
0	234			
1	123	1	?	?
2	111	1	?	?
3	12	9	?	?
4	3	?	?	?

$r_i = r_4 \neq 0$, więc kontynuujemy etap II: $i = 5$, $12 = 4 \cdot 3 + 0$, stąd $r_5 = 0$ i $q_3 = 4$.

Rozszerzony algorytm Euklidesa - przykład

... stąd $r_5 = 0$ i $q_3 = 4$.

i	r_i	q_i	x_i	y_i
0	234			
1	123	1	?	?
2	111	1	?	?
3	12	9	?	?
4	3	4	?	?
5	0	-	-	-

Rozszerzony algorytm Euklidesa - przykład

... stąd $r_5 = 0$ i $q_3 = 4$.

i	r_i	q_i	x_i	y_i
0	234			
1	123	1	?	?
2	111	1	?	?
3	12	9	?	?
4	3	4	?	?
5	0	-	-	-

Rozszerzony algorytm Euklidesa - przykład

W etapie III znów ustawiamy licznik $i = 4$ i wpisujemy $x_i = x_4 = 0$ oraz $y_i = y_4 = 1$.

i	r_i	q_i	x_i	y_i
0	234			
1	123	1	?	?
2	111	1	?	?
3	12	9	?	?
4	3	4	0	1
5	0	-	-	-

Rozszerzony algorytm Euklidesa - przykład

i	r_i	q_i	x_i	y_i
0	234			
1	123	1	?	?
2	111	1	?	?
3	12	9	?	?
4	3	4	0	1
5	0	-	-	-

$i > 1$, więc zmniejszamy licznik pętli o 1 ($i = 3$) i obliczamy:

Rozszerzony algorytm Euklidesa - przykład

i	r_i	q_i	x_i	y_i
0	234			
1	123	1	?	?
2	111	1	?	?
3	12	9	?	?
4	3	4	0	1
5	0	-	-	-

$i > 1$, więc zmniejszamy licznik pętli o 1 ($i = 3$) i obliczamy:

$x_3 := y_4 = 1$;

Rozszerzony algorytm Euklidesa - przykład

i	r_i	q_i	x_i	y_i
0	234			
1	123	1	?	?
2	111	1	?	?
3	12	9	?	?
4	3	4	0	1
5	0	-	-	-

$i > 1$, więc zmniejszamy licznik pętli o 1 ($i = 3$) i obliczamy:
$x_3 := y_4 = 1$; $y_3 := x_4 - q_3 x_3 = 0 - 9 \cdot 1 = -9$.

Rozszerzony algorytm Euklidesa - przykład

...$(i = 3)$ i obliczamy: $x_3 := y_4 = 1$; $y_3 := x_4 - q_3 x_3 = 0 - 9 \cdot 1 = -9$.

i	r_i	q_i	x_i	y_i
0	234			
1	123	1	?	?
2	111	1	?	?
3	12	9	1	-9
4	3	4	0	1
5	0	-	-	-

Rozszerzony algorytm Euklidesa - przykład

i	r_i	q_i	x_i	y_i
0	234			
1	123	1	?	?
2	111	1	?	?
3	12	9	1	-9
4	3	4	0	1
5	0	-	-	-

$i > 1$, więc zmniejszamy licznik pętli o 1 ($i = 2$) i obliczamy:

Rozszerzony algorytm Euklidesa - przykład

i	r_i	q_i	x_i	y_i
0	234			
1	123	1	?	?
2	111	1	?	?
3	12	9	1	-9
4	3	4	0	1
5	0	-	-	-

$i > 1$, więc zmniejszamy licznik pętli o 1 ($i = 2$) i obliczamy:

$x_2 := y_3 = -9$;

Rozszerzony algorytm Euklidesa - przykład

i	r_i	q_i	x_i	y_i
0	234			
1	123	1	?	?
2	111	1	?	?
3	12	9	1	-9
4	3	4	0	1
5	0	-	-	-

$i > 1$, więc zmniejszamy licznik pętli o 1 ($i = 2$) i obliczamy:
$x_2 := y_3 = -9$; $y_2 := x_3 - q_2 x_2 = 1 - 1 \cdot (-9) = 10$.

Rozszerzony algorytm Euklidesa - przykład

...$(i = 2)$ i obliczamy: $x_2 := y_3 = -9$;
$y_2 := x_3 - q_2 x_2 = 1 - 1 \cdot (-9) = 10$.

i	r_i	q_i	x_i	y_i
0	234			
1	123	1	?	?
2	111	1	-9	10
3	12	9	1	-9
4	3	4	0	1
5	0	-	-	-

Rozszerzony algorytm Euklidesa - przykład

i	r_i	q_i	x_i	y_i
0	234			
1	123	1	?	?
2	111	1	-9	10
3	12	9	1	-9
4	3	4	0	1
5	0	-	-	-

$i > 1$, więc zmniejszamy licznik pętli o 1 ($i = 1$) i obliczamy:

Rozszerzony algorytm Euklidesa - przykład

i	r_i	q_i	x_i	y_i
0	234			
1	123	1	?	?
2	111	1	-9	10
3	12	9	1	-9
4	3	4	0	1
5	0	-	-	-

$i > 1$, więc zmniejszamy licznik pętli o 1 ($i = 1$) i obliczamy:

$x_1 := y_2 = 10$;

Rozszerzony algorytm Euklidesa - przykład

i	r_i	q_i	x_i	y_i
0	234			
1	123	1	?	?
2	111	1	-9	10
3	12	9	1	-9
4	3	4	0	1
5	0	-	-	-

$i > 1$, więc zmniejszamy licznik pętli o 1 ($i = 1$) i obliczamy:
$x_1 := y_2 = 10$; $y_1 := x_2 - q_1 x_1 = -9 - 1 \cdot 10 = -19$.

Rozszerzony algorytm Euklidesa - przykład

...$(i = 1)$ i obliczamy: $x_1 := y_2 = 10$;
$y_1 := x_2 - q_1 x_1 = -9 - 1 \cdot 10 = -19$.

i	r_i	q_i	x_i	y_i
0	234			
1	123	1	10	-19
2	111	1	-9	10
3	12	9	1	-9
4	3	4	0	1
5	0	-	-	-

Rozszerzony algorytm Euklidesa - przykład

i	r_i	q_i	x_i	y_i
0	234			
1	123	1	10	-19
2	111	1	-9	10
3	12	9	1	-9
4	3	4	0	1
5	0	-	-	-

Rozszerzony algorytm Euklidesa - przykład

i	r_i	q_i	x_i	y_i
0	234			
1	123	1	10	-19
2	111	1	-9	10
3	12	9	1	-9
4	3	4	0	1
5	0	-	-	-

W tym momencie $i = 1$, więc algorytm się kończy.

Rozszerzony algorytm Euklidesa - przykład

i	r_i	q_i	x_i	y_i
0	234			
1	123	1	10	-19
2	111	1	-9	10
3	12	9	1	-9
4	3	4	0	1
5	0	-	-	-

W tym momencie $i = 1$, więc algorytm się kończy. Z tabeli można spisać wyniki: $NWD(234, 123) = 3$ oraz $234 \cdot 10 + 123 \cdot (-19) = 3$.

Najmniejsza wspólna wielokrotność

Najmniejsza wspólna wielokrotność

Najmniejsza wspólna wielokrotność dodatnich liczb a i b (zapisywana jako NWW(a, b)) to najmniejsza liczba w taka, że $a|w$ i $b|w$.

Najmniejsza wspólna wielokrotność

Najmniejsza wspólna wielokrotność

Najmniejsza wspólna wielokrotność dodatnich liczb a i b (zapisywana jako NWW(a, b)) to najmniejsza liczba w taka, że $a|w$ i $b|w$.

Twierdzonko

Dla dodatnich liczb a, b zachodzi:
$NWW(a, 1) =$

Najmniejsza wspólna wielokrotność

Najmniejsza wspólna wielokrotność

Najmniejsza wspólna wielokrotność dodatnich liczb a i b (zapisywana jako NWW(a, b)) to najmniejsza liczba w taka, że $a|w$ i $b|w$.

Twierdzonko

Dla dodatnich liczb a, b zachodzi:
$NWW(a, 1) = a$, $NWW(a, a) =$

Najmniejsza wspólna wielokrotność

Najmniejsza wspólna wielokrotność

Najmniejsza wspólna wielokrotność dodatnich liczb a i b (zapisywana jako NWW(a, b)) to najmniejsza liczba w taka, że $a|w$ i $b|w$.

Twierdzonko

Dla dodatnich liczb a, b zachodzi:
$NWW(a, 1) = a$, $NWW(a, a) = a$, $NWW(a, b) = NWW(b, a)$.

Najmniejsza wspólna wielokrotność

Najmniejsza wspólna wielokrotność

Najmniejsza wspólna wielokrotność dodatnich liczb a i b (zapisywana jako NWW(a, b)) to najmniejsza liczba w taka, że $a|w$ i $b|w$.

Twierdzonko

Dla dodatnich liczb a, b zachodzi:
$NWW(a, 1) = a$, $NWW(a, a) = a$, $NWW(a, b) = NWW(b, a)$.

Przykłady: $NWW(7, 8) =$

Najmniejsza wspólna wielokrotność

Najmniejsza wspólna wielokrotność

Najmniejsza wspólna wielokrotność dodatnich liczb a i b (zapisywana jako NWW(a, b)) to najmniejsza liczba w taka, że $a|w$ i $b|w$.

Twierdzonko

Dla dodatnich liczb a, b zachodzi:
$NWW(a, 1) = a$, $NWW(a, a) = a$, $NWW(a, b) = NWW(b, a)$.

Przykłady: $NWW(7, 8) = 56$, $NWW(9, 6) =$

Najmniejsza wspólna wielokrotność

Najmniejsza wspólna wielokrotność

Najmniejsza wspólna wielokrotność dodatnich liczb a i b (zapisywana jako NWW(a, b)) to najmniejsza liczba w taka, że $a|w$ i $b|w$.

Twierdzonko

Dla dodatnich liczb a, b zachodzi:
$NWW(a, 1) = a$, $NWW(a, a) = a$, $NWW(a, b) = NWW(b, a)$.

Przykłady: $NWW(7, 8) = 56$, $NWW(9, 6) = 18$.
Jednak dla większych liczb znajdowanie NWW wymaga systematycznego podejścia.

Najmniejsza wspólna wielokrotność - algorytmiczne wyznaczanie

Zależność NWD i NWW

Dla dodatnich liczb a, b zachodzi:

$$a \cdot b = NWD(a, b) \cdot NWW(a, b).$$

W szczególności zachodzi: $NWD(a, b) = \frac{ab}{NWW(a,b)}$ i $NWW(a, b) = \frac{ab}{NWD(a,b)}$.

Najmniejsza wspólna wielokrotność - algorytmiczne wyznaczanie

Zależność NWD i NWW

Dla dodatnich liczb a, b zachodzi:

$$a \cdot b = NWD(a, b) \cdot NWW(a, b).$$

W szczególności zachodzi: $NWD(a, b) = \frac{ab}{NWW(a,b)}$ i $NWW(a, b) = \frac{ab}{NWD(a,b)}$.

Dzięki powyższemu twierdzeniu możemy wyznaczać NWW algorytmicznie: wystarczy wyznaczyć $NWD(a, b)$ z algorytmu Euklidesa i zastosować wzór: $NWW(a, b) = \frac{ab}{NWD(a,b)}$.

Najmniejsza wspólna wielokrotność - podzielność przez wiele liczb

W rozdziale dotyczącym zliczania elementów zbioru (kombinatoryki) przyda się twierdzenie:

Podzielność przez dwie liczby

Dla dodatnich liczb a, b zachodzi:

$$a|c \text{ i } b|c \Leftrightarrow NWW(a,b)|c.$$

Najmniejsza wspólna wielokrotność - podzielność przez wiele liczb

W rozdziale dotyczącym zliczania elementów zbioru (kombinatoryki) przyda się twierdzenie:

Podzielność przez dwie liczby

Dla dodatnich liczb a, b zachodzi:

$$a|c \text{ i } b|c \Leftrightarrow NWW(a,b)|c.$$

Oczywiście, twierdzenie to działa również dla większej ilości liczb niż dwie.

Liczby pierwsze - wstęp

Każda liczba $a > 1$ ma **przynajmniej** 2 dzielniki: 1 i samą siebie.

Liczby pierwsze - wstęp

Każda liczba $a > 1$ ma **przynajmniej** 2 dzielniki: 1 i samą siebie. Jednak niektóre z liczb są pod tym względem wyjątkowe:

Liczba pierwsza

Liczba pierwsza to liczba naturalna posiadająca dokładnie 2 różne dzielniki. Liczbę naturalną większą od 1 nazywamy *złożoną*, gdy nie jest pierwsza.

Liczby pierwsze - wstęp

Każda liczba $a > 1$ ma **przynajmniej** 2 dzielniki: 1 i samą siebie. Jednak niektóre z liczb są pod tym względem wyjątkowe:

Liczba pierwsza

Liczba pierwsza to liczba naturalna posiadająca dokładnie 2 różne dzielniki. Liczbę naturalną większą od 1 nazywamy *złożoną*, gdy nie jest pierwsza.

Liczby 0 i 1 nie są uważane ani za pierwsze, ani za złożone.

Liczby pierwsze - wstęp

Każda liczba $a > 1$ ma **przynajmniej** 2 dzielniki: 1 i samą siebie. Jednak niektóre z liczb są pod tym względem wyjątkowe:

Liczba pierwsza

Liczba pierwsza to liczba naturalna posiadająca dokładnie 2 różne dzielniki. Liczbę naturalną większą od 1 nazywamy *złożoną*, gdy nie jest pierwsza.

Liczby 0 i 1 nie są uważane ani za pierwsze, ani za złożone. Zbiór liczb pierwszych oznaczamy $\mathcal{P}$.

Liczby względnie pierwsze

Jeśli $\mathrm{NWD}(a, b) = 1$ to a i b nazywamy *liczbami względnie pierwszymi*. Zapisujemy $a \perp b$.

Liczby pierwsze - znaczenie

Liczby pierwsze są kluczowe dla teorii liczb, gdyż każdą liczbę można rozłożyć na iloczyn liczb pierwszych w dokładnie jeden sposób (z dokładnością do przestawienia kolejności).

Liczby pierwsze - znaczenie

Liczby pierwsze są kluczowe dla teorii liczb, gdyż każdą liczbę można rozłożyć na iloczyn liczb pierwszych w dokładnie jeden sposób (z dokładnością do przestawienia kolejności). Można to porównać do cząsteczek chemicznych, które może utworzyć tylko jeden układ atomów (choć to porównanie nie jest idealne, bo rozkład na czynniki pierwsze nie zmienia się przy zmianie kolejności tych liczb).

Liczby pierwsze - znaczenie

Liczby pierwsze są kluczowe dla teorii liczb, gdyż każdą liczbę można rozłożyć na iloczyn liczb pierwszych w dokładnie jeden sposób (z dokładnością do przestawienia kolejności). Można to porównać do cząsteczek chemicznych, które może utworzyć tylko jeden układ atomów (choć to porównanie nie jest idealne, bo rozkład na czynniki pierwsze nie zmienia się przy zmianie kolejności tych liczb).

Rozkład (faktoryzacja) liczby całkowitej na czynniki pierwsze

Rozkład liczby całkowitej dodatniej na czynniki pierwsze to zapisanie jej w postaci iloczynu $n = p_1^{a_1} p_2^{a_2} \cdot \ldots \cdot p_k^{a_k}$, gdzie $p_1, \ldots p_k$ są różnymi liczbami pierwszymi, a $a_1, \ldots, a_k$ są liczbami całkowitymi dodatnimi.

Liczby pierwsze - znaczenie

Liczby pierwsze są kluczowe dla teorii liczb, gdyż każdą liczbę można rozłożyć na iloczyn liczb pierwszych w dokładnie jeden sposób (z dokładnością do przestawienia kolejności). Można to porównać do cząsteczek chemicznych, które może utworzyć tylko jeden układ atomów (choć to porównanie nie jest idealne, bo rozkład na czynniki pierwsze nie zmienia się przy zmianie kolejności tych liczb).

Rozkład (faktoryzacja) liczby całkowitej na czynniki pierwsze

Rozkład liczby całkowitej dodatniej na czynniki pierwsze to zapisanie jej w postaci iloczynu $n = p_1^{a_1} p_2^{a_2} \cdot \ldots \cdot p_k^{a_k}$, gdzie $p_1, \ldots p_k$ są różnymi liczbami pierwszymi, a $a_1, \ldots, a_k$ są liczbami całkowitymi dodatnimi.

$$600 = 2^3 \cdot 3 \cdot 5^2 = 5^2 \cdot 2^3 \cdot 3 \ldots$$

Fundamentalne twierdzenie arytmetyki

Fundamentalne twierdzenie arytmetyki

Każda liczba naturalna $n > 1$ ma jednoznaczny (z dokładnością do kolejności mnożenia) rozkład (czyli faktoryzację) na iloczyn liczb pierwszych.

Fundamentalne twierdzenie arytmetyki

Fundamentalne twierdzenie arytmetyki

Każda liczba naturalna $n > 1$ ma jednoznaczny (z dokładnością do kolejności mnożenia) rozkład (czyli faktoryzację) na iloczyn liczb pierwszych.

Najważniejszy dla informatyki jest fakt, że to twierdzenie nie jest w żaden sposób konstruktywne tj. nie mówi, jak ten rozkład można wyliczyć.

Fundamentalne twierdzenie arytmetyki

Fundamentalne twierdzenie arytmetyki

Każda liczba naturalna $n > 1$ ma jednoznaczny (z dokładnością do kolejności mnożenia) rozkład (czyli faktoryzację) na iloczyn liczb pierwszych.

Najważniejszy dla informatyki jest fakt, że to twierdzenie nie jest w żaden sposób konstruktywne tj. nie mówi, jak ten rozkład można wyliczyć. Obecnie nie jest znany żaden efektywny algorytm faktoryzujący liczby naturalne, tzn. znajdujący rozkład na iloczyn liczb pierwszych, co jest sednem współczesnych systemów kryptograficznych i innych „algorytmów zapadkowych".

Fundamentalne twierdzenie arytmetyki

Fundamentalne twierdzenie arytmetyki

Każda liczba naturalna $n > 1$ ma jednoznaczny (z dokładnością do kolejności mnożenia) rozkład (czyli faktoryzację) na iloczyn liczb pierwszych.

Najważniejszy dla informatyki jest fakt, że to twierdzenie nie jest w żaden sposób konstruktywne tj. nie mówi, jak ten rozkład można wyliczyć. Obecnie nie jest znany żaden efektywny algorytm faktoryzujący liczby naturalne, tzn. znajdujący rozkład na iloczyn liczb pierwszych, co jest sednem współczesnych systemów kryptograficznych i innych „algorytmów zapadkowych". Oczywiście, niektóre liczby można rozłożyć stosunkowo łatwo - najtrudniejsze do rozłożenia zaś są te, które są iloczynami dwu liczb pierwszych podobnej wielkości - za znalezienie takich rozkładów wiele firm wypłaca wysokie nagrody.

Fundamentalne twierdzenie arytmetyki - dodatkowe uwagi

Fundamentalne twierdzenie arytmetyki - dodatkowe uwagi

- Rozkład liczb a i b na czynniki pierwsze automatycznie zadaje nam ich NWD (wystarczy wymnożyć te czynniki pierwsze, które pojawiają się w obydwu rozkładach). Jednak znajdowanie NWD algorytmem Euklidesa jest zazwyczaj o wiele efektywniejsze.

Fundamentalne twierdzenie arytmetyki - dodatkowe uwagi

- Rozkład liczb a i b na czynniki pierwsze automatycznie zadaje nam ich NWD (wystarczy wymnożyć te czynniki pierwsze, które pojawiają się w obydwu rozkładach). Jednak znajdowanie NWD algorytmem Euklidesa jest zazwyczaj o wiele efektywniejsze.
- Choć rozkład liczby na czynniki pierwsze jest algorytmicznie nieosiągalny w sensownym czasie, to sprawdzenie, czy jakaś liczba jest pierwsza jest dużo prostsze: istnieją algorytmy sprawdzające to w czasie „logarytmicznym" - czyli dość szybko (ważne dla „algorytmów zapadkowych").

Twierdzenie o liczbach pierwszych

Nie istnieje „największa” liczba pierwsza, gdyż zachodzi twierdzenie (również udowodnione przez Euklidesa):

Twierdzenie o liczbach pierwszych

Liczb pierwszych jest nieskończenie wiele.

Twierdzenie o liczbach pierwszych

Nie istnieje „największa” liczba pierwsza, gdyż zachodzi twierdzenie (również udowodnione przez Euklidesa):

Twierdzenie o liczbach pierwszych

Liczb pierwszych jest nieskończenie wiele.

Dowód tego twierdzenia będzie zapewne jedynym dowodem w ramach tego kursu wymaganym na egzaminie (bo jest ładny).

Dowód twierdzenia o liczbach pierwszych

Twierdzenie o liczbach pierwszych

Liczb pierwszych jest nieskończenie wiele.

Dowód twierdzenia o liczbach pierwszych

Twierdzenie o liczbach pierwszych

Liczb pierwszych jest nieskończenie wiele.

Dowód przeprowadzamy metodą „nie wprost”.

Dowód twierdzenia o liczbach pierwszych

Twierdzenie o liczbach pierwszych

Liczb pierwszych jest nieskończenie wiele.

Dowód przeprowadzamy metodą „nie wprost". Zakładamy, że liczb pierwszych jest skończenie wiele i są to: $p_1, \ldots, p_k$.

Dowód twierdzenia o liczbach pierwszych

Twierdzenie o liczbach pierwszych

Liczb pierwszych jest nieskończenie wiele.

Dowód przeprowadzamy metodą „nie wprost". Zakładamy, że liczb pierwszych jest skończenie wiele i są to: $p_1, \ldots, p_k$. Rozważmy liczbę $n = p_1 p_2 \ldots p_k + 1$.

Dowód twierdzenia o liczbach pierwszych

Twierdzenie o liczbach pierwszych

Liczb pierwszych jest nieskończenie wiele.

Dowód przeprowadzamy metodą „nie wprost". Zakładamy, że liczb pierwszych jest skończenie wiele i są to: $p_1, \ldots, p_k$. Rozważmy liczbę $n = p_1 p_2 \ldots p_k + 1$. Jest ona oczywiście większa od każdej z liczb $p_1, \ldots, p_k$.

Dowód twierdzenia o liczbach pierwszych

Twierdzenie o liczbach pierwszych

Liczb pierwszych jest nieskończenie wiele.

Dowód przeprowadzamy metodą „nie wprost". Zakładamy, że liczb pierwszych jest skończenie wiele i są to: $p_1, \ldots, p_k$. Rozważmy liczbę $n = p_1 p_2 \ldots p_k + 1$. Jest ona oczywiście większa od każdej z liczb $p_1, \ldots, p_k$. Ponadto żadna z liczb pierwszych p_i nie dzieli n,

Dowód twierdzenia o liczbach pierwszych

Twierdzenie o liczbach pierwszych

Liczb pierwszych jest nieskończenie wiele.

Dowód przeprowadzamy metodą „nie wprost". Zakładamy, że liczb pierwszych jest skończenie wiele i są to: $p_1, \ldots, p_k$. Rozważmy liczbę $n = p_1 p_2 \ldots p_k + 1$. Jest ona oczywiście większa od każdej z liczb $p_1, \ldots, p_k$. Ponadto żadna z liczb pierwszych p_i nie dzieli n, bo n przy dzieleniu przez każde p_i daje resztę 1.

Dowód twierdzenia o liczbach pierwszych

Twierdzenie o liczbach pierwszych

Liczb pierwszych jest nieskończenie wiele.

Dowód przeprowadzamy metodą „nie wprost". Zakładamy, że liczb pierwszych jest skończenie wiele i są to: $p_1, \ldots, p_k$. Rozważmy liczbę $n = p_1 p_2 \ldots p_k + 1$. Jest ona oczywiście większa od każdej z liczb $p_1, \ldots, p_k$. Ponadto żadna z liczb pierwszych p_i nie dzieli n, bo n przy dzieleniu przez każde p_i daje resztę 1. n, jako liczba całkowita, musi mieć jednoznaczny rozkład na czynniki pierwsze.

Dowód twierdzenia o liczbach pierwszych

Twierdzenie o liczbach pierwszych

Liczb pierwszych jest nieskończenie wiele.

Dowód przeprowadzamy metodą „nie wprost". Zakładamy, że liczb pierwszych jest skończenie wiele i są to: $p_1, \ldots, p_k$. Rozważmy liczbę $n = p_1 p_2 \ldots p_k + 1$. Jest ona oczywiście większa od każdej z liczb $p_1, \ldots, p_k$. Ponadto żadna z liczb pierwszych p_i nie dzieli n, bo n przy dzieleniu przez każde p_i daje resztę 1. n, jako liczba całkowita, musi mieć jednoznaczny rozkład na czynniki pierwsze. A zatem są tylko dwie możliwości: w rozkładzie n są liczby pierwsze, których do tej pory nie wymieniliśmy (czyli inne niż $p_1, \ldots, p_k$), lub też samo n jest nową liczbą pierwszą (także różną od dotychczasowych).

Dowód twierdzenia o liczbach pierwszych

Twierdzenie o liczbach pierwszych

Liczb pierwszych jest nieskończenie wiele.

Dowód przeprowadzamy metodą „nie wprost". Zakładamy, że liczb pierwszych jest skończenie wiele i są to: $p_1, \ldots, p_k$. Rozważmy liczbę $n = p_1 p_2 \ldots p_k + 1$. Jest ona oczywiście większa od każdej z liczb $p_1, \ldots, p_k$. Ponadto żadna z liczb pierwszych p_i nie dzieli n, bo n przy dzieleniu przez każde p_i daje resztę 1. n, jako liczba całkowita, musi mieć jednoznaczny rozkład na czynniki pierwsze. A zatem są tylko dwie możliwości: w rozkładzie n są liczby pierwsze, których do tej pory nie wymieniliśmy (czyli inne niż $p_1, \ldots, p_k$), lub też samo n jest nową liczbą pierwszą (także różną od dotychczasowych). Obydwie te możliwości są sprzeczne z założeniem dowodu nie wprost, więc to założenie (o skończonej ilości liczb pierwszych) musi być fałszywe. QED.

Największa znana liczba pierwsza

Mimo to, dla kryptografów istotna jest znajomość jak największych liczb pierwszych.

Największa znana liczba pierwsza

Mimo to, dla kryptografów istotna jest znajomość jak największych liczb pierwszych. Największą **znaną** (tj. taką, której pierwszość udowodniono) liczbą pierwszą (na moment rozpoczęcia tego semestru tj. 1 X 2021) jest $2^{82589933} - 1$. Liczy sobie 24862048 cyfr w zapisie dziesiętnym. Jej pierwszość udowodnił w grudniu 2018 roku Patrick Laroche.

Arytmetyka modularna - motywacja

Zauważmy, że do modelowania niektórych zjawisk (i opisywania ich algorytmami) zwykła arytmetyka liczb naturalnych nie jest wystarczająca. Dotyczy to szczególnie zjawisk cyklicznych jak np. dzienna rachuba czasu.

Arytmetyka modularna - motywacja

Zauważmy, że do modelowania niektórych zjawisk (i opisywania ich algorytmami) zwykła arytmetyka liczb naturalnych nie jest wystarczająca. Dotyczy to szczególnie zjawisk cyklicznych jak np. dzienna rachuba czasu. Jeśli jest godzina 3 w nocy, to odpowiadając na pytanie, która godzina była 5 godzin temu nie mówimy, że „minus druga" $(3 - 5)$ tylko 22.

Arytmetyka modularna - motywacja

Zauważmy, że do modelowania niektórych zjawisk (i opisywania ich algorytmami) zwykła arytmetyka liczb naturalnych nie jest wystarczająca. Dotyczy to szczególnie zjawisk cyklicznych jak np. dzienna rachuba czasu. Jeśli jest godzina 3 w nocy, to odpowiadając na pytanie, która godzina była 5 godzin temu nie mówimy, że „minus druga" $(3-5)$ tylko 22. Analogicznie, jeśli zaśniemy o 21 na 9 godzin, to wiemy, że zbudzimy się o 6, a nie o 30 $(21+9)$.

Arytmetyka modularna - motywacja

Zauważmy, że do modelowania niektórych zjawisk (i opisywania ich algorytmami) zwykła arytmetyka liczb naturalnych nie jest wystarczająca. Dotyczy to szczególnie zjawisk cyklicznych jak np. dzienna rachuba czasu. Jeśli jest godzina 3 w nocy, to odpowiadając na pytanie, która godzina była 5 godzin temu nie mówimy, że „minus druga" $(3-5)$ tylko 22. Analogicznie, jeśli zaśniemy o 21 na 9 godzin, to wiemy, że zbudzimy się o 6, a nie o 30 $(21+9)$. W tym przykładzie nie interesuje nas właściwy wynik, lecz jego reszta z dzielenia przez 24.

Arytmetyka modularna - motywacja

Zauważmy, że do modelowania niektórych zjawisk (i opisywania ich algorytmami) zwykła arytmetyka liczb naturalnych nie jest wystarczająca. Dotyczy to szczególnie zjawisk cyklicznych jak np. dzienna rachuba czasu. Jeśli jest godzina 3 w nocy, to odpowiadając na pytanie, która godzina była 5 godzin temu nie mówimy, że „minus druga" $(3-5)$ tylko 22. Analogicznie, jeśli zaśniemy o 21 na 9 godzin, to wiemy, że zbudzimy się o 6, a nie o 30 $(21+9)$. W tym przykładzie nie interesuje nas właściwy wynik, lecz jego reszta z dzielenia przez 24.
Takie i inne cykliczne zjawiska (np. zmiany pór roku, względne przemieszczenia mechanizmu złożonego z kół zębatych itp.) modeluje arytmetyka modularna.

Przystawanie modulo

Przystawanie modulo

Mówimy, że dwie liczby a i b *przystają do siebie modulo* n, jeśli ich różnica $a - b$ jest wielokrotnością n (lub innymi słowy, jeśli liczby te dają tę samą resztę z dzielenia przez n). Zapisujemy to symbolem $a \equiv b(mod\ n)$ lub $a \equiv_n b$.

Przystawanie modulo

Przystawanie modulo

Mówimy, że dwie liczby a i b *przystają do siebie modulo* n, jeśli ich różnica $a - b$ jest wielokrotnością n (lub innymi słowy, jeśli liczby te dają tę samą resztę z dzielenia przez n). Zapisujemy to symbolem $a \equiv b(mod\ n)$ lub $a \equiv_n b$.

Nieprzypadkowo oznaczenie jest tak podobne do oznaczenia reszty z dzielenia. Jeśli $a = b \mod n$ to $a \equiv b(mod\ n)$

Przystawanie modulo

Przystawanie modulo

Mówimy, że dwie liczby a i b *przystają do siebie modulo* n, jeśli ich różnica $a - b$ jest wielokrotnością n (lub innymi słowy, jeśli liczby te dają tę samą resztę z dzielenia przez n). Zapisujemy to symbolem $a \equiv b(mod\ n)$ lub $a \equiv_n b$.

Nieprzypadkowo oznaczenie jest tak podobne do oznaczenia reszty z dzielenia. Jeśli $a = b \mod n$ to $a \equiv b(mod\ n)$ (niekoniecznie na odwrót, bo w pierwszym przypadku musi zachodzić $a < n$, a w drugim niekoniecznie - np. $8 \equiv 13(mod\ 5)$, choć resztą z dzielenia 13 przez 5 jest 3).

Przystawanie modulo - równoważność

Łatwo sprawdzić, że relacja przystawania modulo jest równoważnością na zbiorze liczb całkowitych, czyli

Równoważność przystawania modulo

Dla dowolnych a, b, c oraz $n > 0$ mamy:

a) $a \equiv_n a$,

b) $a \equiv_n b \Leftrightarrow b \equiv_n a$,

c) jeśli $a \equiv_n b$ i $b \equiv_n c$, to $a \equiv_n c$.

Przystawanie modulo - równoważność

Łatwo sprawdzić, że relacja przystawania modulo jest równoważnością na zbiorze liczb całkowitych, czyli

Równoważność przystawania modulo

Dla dowolnych a, b, c oraz $n > 0$ mamy:
a) $a \equiv_n a$,
b) $a \equiv_n b \Leftrightarrow b \equiv_n a$,
c) jeśli $a \equiv_n b$ i $b \equiv_n c$, to $a \equiv_n c$.

Klasami abstrakcji tej relacji są zbiory liczb dające tę samą resztę z dzielenia przez n - czyli w skrócie reszty z dzielenia przez n.

Przystawanie modulo - własności

Ponadto spełnione są własności:

Własności przystawania modulo

Dla dowolnych a, b, c, d oraz $n > 0$ mamy:
a) jeśli $a \equiv_n b$ i $c \equiv_n d$, to $a + c \equiv_n b + d$,
b) jeśli $a \equiv_n b$ i $c \equiv_n d$, to $a - c \equiv_n b - d$,
c) jeśli $a \equiv_n b$ i $c \equiv_n d$, to $ac \equiv_n bd$.

Przystawanie modulo - własności

Ponadto spełnione są własności:

Własności przystawania modulo

Dla dowolnych a, b, c, d oraz $n > 0$ mamy:
a) jeśli $a \equiv_n b$ i $c \equiv_n d$, to $a + c \equiv_n b + d$,
b) jeśli $a \equiv_n b$ i $c \equiv_n d$, to $a - c \equiv_n b - d$,
c) jeśli $a \equiv_n b$ i $c \equiv_n d$, to $ac \equiv_n bd$.

Np. Niech $a \equiv_{17} 5$ i $b \equiv_{17} 3$.

Przystawanie modulo - własności

Ponadto spełnione są własności:

Własności przystawania modulo

Dla dowolnych a, b, c, d oraz $n > 0$ mamy:
a) jeśli $a \equiv_n b$ i $c \equiv_n d$, to $a + c \equiv_n b + d$,
b) jeśli $a \equiv_n b$ i $c \equiv_n d$, to $a - c \equiv_n b - d$,
c) jeśli $a \equiv_n b$ i $c \equiv_n d$, to $ac \equiv_n bd$.

Np. Niech $a \equiv_{17} 5$ i $b \equiv_{17} 3$. Wtedy $a + b \equiv_{17}$

Przystawanie modulo - własności

Ponadto spełnione są własności:

Własności przystawania modulo

Dla dowolnych a, b, c, d oraz $n > 0$ mamy:
a) jeśli $a \equiv_n b$ i $c \equiv_n d$, to $a + c \equiv_n b + d$,
b) jeśli $a \equiv_n b$ i $c \equiv_n d$, to $a - c \equiv_n b - d$,
c) jeśli $a \equiv_n b$ i $c \equiv_n d$, to $ac \equiv_n bd$.

Np. Niech $a \equiv_{17} 5$ i $b \equiv_{17} 3$. Wtedy $a + b \equiv_{17} 8$, $a - b \equiv_{17}$

Przystawanie modulo - własności

Ponadto spełnione są własności:

Własności przystawania modulo

Dla dowolnych a, b, c, d oraz $n > 0$ mamy:
a) jeśli $a \equiv_n b$ i $c \equiv_n d$, to $a + c \equiv_n b + d$,
b) jeśli $a \equiv_n b$ i $c \equiv_n d$, to $a - c \equiv_n b - d$,
c) jeśli $a \equiv_n b$ i $c \equiv_n d$, to $ac \equiv_n bd$.

Np. Niech $a \equiv_{17} 5$ i $b \equiv_{17} 3$. Wtedy $a + b \equiv_{17} 8$, $a - b \equiv_{17} 2$, $ab \equiv_{17}$

Przystawanie modulo - własności

Ponadto spełnione są własności:

Własności przystawania modulo

Dla dowolnych a, b, c, d oraz $n > 0$ mamy:
a) jeśli $a \equiv_n b$ i $c \equiv_n d$, to $a + c \equiv_n b + d$,
b) jeśli $a \equiv_n b$ i $c \equiv_n d$, to $a - c \equiv_n b - d$,
c) jeśli $a \equiv_n b$ i $c \equiv_n d$, to $ac \equiv_n bd$.

Np. Niech $a \equiv_{17} 5$ i $b \equiv_{17} 3$. Wtedy $a + b \equiv_{17} 8$, $a - b \equiv_{17} 2$, $ab \equiv_{17} 15$.

Arytmetyka modularna

Dzięki tym własnościom, można zdefiniować tzw. *kongruencje* czyli działania na klasach abstrakcji relacji modulo (czyli na resztach z dzielenia) tak samo jak na liczbach.

Arytmetyka modularna

Dzięki tym własnościom, można zdefiniować tzw. *kongruencje* czyli działania na klasach abstrakcji relacji modulo (czyli na resztach z dzielenia) tak samo jak na liczbach.
$3 + 5 \equiv_6$

Arytmetyka modularna

Dzięki tym własnościom, można zdefiniować tzw. *kongruencje* czyli działania na klasach abstrakcji relacji modulo (czyli na resztach z dzielenia) tak samo jak na liczbach.
$3 + 5 \equiv_6 2$, $3 - 5 \equiv_6$

Arytmetyka modularna

Dzięki tym własnościom, można zdefiniować tzw. *kongruencje* czyli działania na klasach abstrakcji relacji modulo (czyli na resztach z dzielenia) tak samo jak na liczbach.
$3 + 5 \equiv_6 2$, $3 - 5 \equiv_6 4$, $3 \cdot 5 \equiv_6$

Arytmetyka modularna

Dzięki tym własnościom, można zdefiniować tzw. *kongruencje* czyli działania na klasach abstrakcji relacji modulo (czyli na resztach z dzielenia) tak samo jak na liczbach.
$3 + 5 \equiv_6 2$, $3 - 5 \equiv_6 4$, $3 \cdot 5 \equiv_6 3$.

Arytmetyka modularna

Dzięki tym własnościom, można zdefiniować tzw. *kongruencje* czyli działania na klasach abstrakcji relacji modulo (czyli na resztach z dzielenia) tak samo jak na liczbach.
$3 + 5 \equiv_6 2$, $3 - 5 \equiv_6 4$, $3 \cdot 5 \equiv_6 3$.
Przez $\mathbb{Z}_n$ będziemy oznaczać zbiór reszt z dzielenia przez n z działaniami arytmetycznymi modulo n.

Arytmetyka modularna - przykład $\mathbb{Z}_4$

Oto „tabliczka dodawania” w zbiorze $\mathbb{Z}_4$:

$+$	0	1	2	3
0	0	1	2	3
1	1	2	3	0
2	2	3	0	1
3	3	0	1	2

Arytmetyka modularna - przykład $\mathbb{Z}_4$

Oto „tabliczka dodawania” w zbiorze $\mathbb{Z}_4$:

$+$	0	1	2	3
0	0	1	2	3
1	1	2	3	0
2	2	3	0	1
3	3	0	1	2

Arytmetyka modularna - przykład $\mathbb{Z}_4$

Oto „tabliczka dodawania” w zbiorze $\mathbb{Z}_4$:

$+$	0	1	2	3
0	0	1	2	3
1	1	2	3	0
2	2	3	0	1
3	3	0	1	2

I tabliczka mnożenia w tym samym zbiorze:

$\cdot$	0	1	2	3
0	0	0	0	0
1	0	1	2	3
2	0	2	0	2
3	0	3	2	1

Arytmetyka modularna - przykład $\mathbb{Z}_4$

Oto „tabliczka dodawania" w zbiorze $\mathbb{Z}_4$:

$+$	0	1	2	3
0	0	1	2	3
1	1	2	3	0
2	2	3	0	1
3	3	0	1	2

I tabliczka mnożenia w tym samym zbiorze:

$\cdot$	0	1	2	3
0	0	0	0	0
1	0	1	2	3
2	0	2	0	2
3	0	3	2	1

Zauważmy, że o ile łatwo definiować odejmowanie za pomocą dodawania, to dzielenie modulo trzeba wykonywać ostrożnie, bo czasem nie ma sensu (np. $3 : 2$ w $\mathbb{Z}_4$), a czasem może dawać wiele wyników (np. $2 : 2$ w $\mathbb{Z}_4$).

Reguła skracania

Reguła skracania

Dla $n > 0$ jeśli $ad \equiv bd (mod\ n)$ i $d \perp n$ to $a \equiv b (mod\ n)$.

Reguła skracania

Reguła skracania

Dla $n > 0$ jeśli $ad \equiv bd (mod\ n)$ i $d \perp n$ to $a \equiv b (mod\ n)$.

Reguła ta mówi, kiedy możemy legalnie wykonać „dzielenie stronami” w arytmetyce modulo n - otóż tylko wtedy, gdy n i liczba przez którą dzielimy są względnie pierwsze.

Reguła skracania

Reguła skracania

Dla $n > 0$ jeśli $ad \equiv bd (mod\ n)$ i $d \perp n$ to $a \equiv b (mod\ n)$.

Reguła ta mówi, kiedy możemy legalnie wykonać „dzielenie stronami" w arytmetyce modulo n - otóż tylko wtedy, gdy n i liczba przez którą dzielimy są względnie pierwsze.
Przykładowo, z faktu, że $4 \cdot 4 \equiv_8 2 \cdot 4 (\equiv_8 0)$ nie wynika, że $4 \equiv_8 2$, bo 4 i 8 nie są względnie pierwsze.

Reguła skracania

Reguła skracania

Dla $n > 0$ jeśli $ad \equiv bd (mod\ n)$ i $d \perp n$ to $a \equiv b (mod\ n)$.

Reguła ta mówi, kiedy możemy legalnie wykonać „dzielenie stronami" w arytmetyce modulo n - otóż tylko wtedy, gdy n i liczba przez którą dzielimy są względnie pierwsze.
Przykładowo, z faktu, że $4 \cdot 4 \equiv_8 2 \cdot 4 (\equiv_8 0)$ nie wynika, że $4 \equiv_8 2$, bo 4 i 8 nie są względnie pierwsze.
Dlatego we wszelkich obliczeniach najlepiej unikać „dzielenia stronami", chyba, że jesteśmy pewni, że działa (tj. sprawdzimy, że liczba przez którą dzielimy i podstawa działania modulo są względnie pierwsze).

Kongruencje liniowe i ich układy

Kongruencją liniową nazywamy przystawanie postaci $ax \equiv_n b$, gdzie $a, b \in \mathbb{Z}$, zaś $x \in \mathbb{Z}_n$ jest niewiadomą. Jeśli kongruencja ta jest prawdziwa dla pewnego x, to x nazywamy *rozwiązaniem* tej kongruencji.

Kongruencje liniowe i ich układy

Kongruencją liniową nazywamy przystawanie postaci $ax \equiv_n b$, gdzie $a, b \in \mathbb{Z}$, zaś $x \in \mathbb{Z}_n$ jest niewiadomą. Jeśli kongruencja ta jest prawdziwa dla pewnego x, to x nazywamy *rozwiązaniem* tej kongruencji.

Układem k kongruencji liniowych będziemy nazywać zbiór k przystawań postaci $Ax \equiv_n b$, gdzie A jest macierzą $k \times k$ o współczynnikach z $\mathbb{Z}$, $b \in \mathbb{Z}^k$ jest wektorem złożonym z k liczb całkowitych, zaś $x \in \mathbb{Z}_n^k$ jest wektorem niewiadomych. Jeśli ten układ kongruencji jest prawdziwy dla pewnego x, to x nazywamy *rozwiązaniem* tej kongruencji.

Kongruencje liniowe i ich układy

Kongruencją liniową nazywamy przystawanie postaci $ax \equiv_n b$, gdzie $a, b \in \mathbb{Z}$, zaś $x \in \mathbb{Z}_n$ jest niewiadomą. Jeśli kongruencja ta jest prawdziwa dla pewnego x, to x nazywamy *rozwiązaniem* tej kongruencji.

Układem k kongruencji liniowych będziemy nazywać zbiór k przystawań postaci $Ax \equiv_n b$, gdzie A jest macierzą $k \times k$ o współczynnikach z $\mathbb{Z}$, $b \in \mathbb{Z}^k$ jest wektorem złożonym z k liczb całkowitych, zaś $x \in \mathbb{Z}_n^k$ jest wektorem niewiadomych. Jeśli ten układ kongruencji jest prawdziwy dla pewnego x, to x nazywamy *rozwiązaniem* tej kongruencji.

Rozwiązywanie kongruencji i ich układów

Kongruencje liniowe i ich układy generalnie rozwiązujemy podobnie jak równości i ich układy na liczbach rzeczywistych. Jest jednak kilka wyjątków:

Rozwiązywanie kongruencji i ich układów

Kongruencje liniowe i ich układy generalnie rozwiązujemy podobnie jak równości i ich układy na liczbach rzeczywistych. Jest jednak kilka wyjątków:

- Dla kongruencji modulo n używamy, zarówno w wyniku jak i w trakcie obliczeń, tylko liczb naturalnych od 0 do $n-1$. Ostatecznie można dopuścić pojawienie się w kongruencji liczb całkowitych większych na moduł od n, lecz powinno się je jak najszybciej dodawać lub odejmować od tych liczb odpowiednie wielokrotności n, by wrócić do właściwego przedziału.

Rozwiązywanie kongruencji i ich układów

Kongruencje liniowe i ich układy generalnie rozwiązujemy podobnie jak równości i ich układy na liczbach rzeczywistych. Jest jednak kilka wyjątków:

- Dla kongruencji modulo n używamy, zarówno w wyniku jak i w trakcie obliczeń, tylko liczb naturalnych od 0 do $n-1$. Ostatecznie można dopuścić pojawienie się w kongruencji liczb całkowitych większych na moduł od n, lecz powinno się je jak najszybciej dodawać lub odejmować od tych liczb odpowiednie wielokrotności n, by wrócić do właściwego przedziału.

Rozwiązywanie kongruencji i ich układów

Kongruencje liniowe i ich układy generalnie rozwiązujemy podobnie jak równości i ich układy na liczbach rzeczywistych. Jest jednak kilka wyjątków:

- Dla kongruencji modulo n używamy, zarówno w wyniku jak i w trakcie obliczeń, tylko liczb naturalnych od 0 do $n-1$. Ostatecznie można dopuścić pojawienie się w kongruencji liczb całkowitych większych na moduł od n, lecz powinno się je jak najszybciej dodawać lub odejmować od tych liczb odpowiednie wielokrotności n, by wrócić do właściwego przedziału.
- Zgodnie z regułą skracania, kongruencje modulo n można dzielić stronami tylko przez liczby względnie pierwsze z n.

Rozwiązywanie kongruencji i ich układów

Kongruencje i ich układy generalnie rozwiązujemy tak jak równości i ich układy na liczbach rzeczywistych. Jest jednak kilka wyjątków:

Rozwiązywanie kongruencji i ich układów

Kongruencje i ich układy generalnie rozwiązujemy tak jak równości i ich układy na liczbach rzeczywistych. Jest jednak kilka wyjątków:

- Z reguły skracania ponadto wynika, że podczas rozwiązywania kongruencji, nie powinno się również mnożyć stronami przez liczby, które nie są względnie pierwsze z n. Co prawda, rozwiązania oryginalnej kongruencji są rozwiązaniami kongruencji przemnożonej stronami przez dowolną liczbę, ale nie jest na odwrót: możemy uzyskać nowe rozwiązania, które nie były rozwiązaniami kongruencji startowej.

Rozwiązywanie kongruencji i ich układów

Kongruencje i ich układy generalnie rozwiązujemy tak jak równości i ich układy na liczbach rzeczywistych. Jest jednak kilka wyjątków:

- Z reguły skracania ponadto wynika, że podczas rozwiązywania kongruencji, nie powinno się również mnożyć stronami przez liczby, które nie są względnie pierwsze z n. Co prawda, rozwiązania oryginalnej kongruencji są rozwiązaniami kongruencji przemnożonej stronami przez dowolną liczbę, ale nie jest na odwrót: możemy uzyskać nowe rozwiązania, które nie były rozwiązaniami kongruencji startowej.

Przykład: $3x \equiv_4 3$. Oczywiście jedynym rozwiązaniem jest $x \equiv_4 1$ (zgodnie z regułą skracania, możemy kongruencję modulo 4 podzielić stronami przez 3). Gdybyśmy pomnożyli kongruencję stronami przez 2 (które nie jest względnie pierwsze z 4), otrzymalibyśmy $6x \equiv_4 6 \Leftrightarrow 2x \equiv_4 2$, co ma dwa rozwiązania $x \equiv_4 1$ i $x \equiv_4 3$. To drugie nie jest rozwiązaniem wyjściowej kongruencji.

Kongruencja liniowa - przykład

Zadanie

Rozwiązać kongruencję

$$7x \equiv_{10} 6$$

Kongruencja liniowa - przykład

Zadanie

Rozwiązać kongruencję

$$7x \equiv_{10} 6$$

Jak widać, operujemy działaniami w zbiorze $\mathbb{Z}_{10}$ - pamiętajmy więc, że mamy do dyspozycji tylko liczby całkowite od 0 do 9 i nie możemy używać ułamków!

Kongruencja liniowa - przykład

Zadanie

Rozwiązać kongruencję

$$7x \equiv_{10} 6$$

Jak widać, operujemy działaniami w zbiorze $\mathbb{Z}_{10}$ - pamiętajmy więc, że mamy do dyspozycji tylko liczby całkowite od 0 do 9 i nie możemy używać ułamków! Aby rozwiązać to równanie, wystarczy podzielić obie strony przez 7. Czy możemy to zrobić?

Kongruencja liniowa - przykład

Zadanie

Rozwiązać kongruencję

$$7x \equiv_{10} 6$$

Jak widać, operujemy działaniami w zbiorze $\mathbb{Z}_{10}$ - pamiętajmy więc, że mamy do dyspozycji tylko liczby całkowite od 0 do 9 i nie możemy używać ułamków! Aby rozwiązać to równanie, wystarczy podzielić obie strony przez 7. Czy możemy to zrobić? Zgodnie z regułą skracania tak, bo $7 \perp 10$!

Kongruencja liniowa - przykład

Zadanie

$$7x \equiv_{10} 6$$

Kongruencja liniowa - przykład

Zadanie

$$7x \equiv_{10} 6$$

By wykonać dzielenie 6 przez 7 modulo 10, szukam liczby równoważnej 6 w tej arytmetyce, którą potrafię podzielić przez 7.

Kongruencja liniowa - przykład

Zadanie

$$7x \equiv_{10} 6$$

By wykonać dzielenie 6 przez 7 modulo 10, szukam liczby równoważnej 6 w tej arytmetyce, którą potrafię podzielić przez 7. Sprawdzam kolejne liczby:

Kongruencja liniowa - przykład

Zadanie

$$7x \equiv_{10} 6$$

By wykonać dzielenie 6 przez 7 modulo 10, szukam liczby równoważnej 6 w tej arytmetyce, którą potrafię podzielić przez 7. Sprawdzam kolejne liczby: nie może to być 16, 26, 36, 46, bo te liczby, choć równoważne 6 nie dzielą się przez 7.

Kongruencja liniowa - przykład

Zadanie

$$7x \equiv_{10} 6$$

By wykonać dzielenie 6 przez 7 modulo 10, szukam liczby równoważnej 6 w tej arytmetyce, którą potrafię podzielić przez 7. Sprawdzam kolejne liczby: nie może to być 16, 26, 36, 46, bo te liczby, choć równoważne 6 nie dzielą się przez 7. Będzie to np. 56 , stąd $7 \cdot 8 \equiv_{10} 6$, więc $6 : 7 \equiv_{10} 8$.

Kongruencja liniowa - przykład

Zadanie

$$7x \equiv_{10} 6$$

By wykonać dzielenie 6 przez 7 modulo 10, szukam liczby równoważnej 6 w tej arytmetyce, którą potrafię podzielić przez 7. Sprawdzam kolejne liczby: nie może to być 16, 26, 36, 46, bo te liczby, choć równoważne 6 nie dzielą się przez 7. Będzie to np. 56 , stąd $7 \cdot 8 \equiv_{10} 6$, więc $6 : 7 \equiv_{10} 8$. Stąd $x \equiv_{10} 8$ jest rozwiązaniem tej kongruencji.

Układ kongruencji - przykład

Zadanie z egzaminu (2014, zaoczne, I termin)

$$\begin{cases} 3x - 5y \equiv_{13} 1 \\ 9x - 4y \equiv_{13} 10. \end{cases}$$

Układ kongruencji - przykład

Zadanie z egzaminu (2014, zaoczne, I termin)

$$\begin{cases} 3x - 5y \equiv_{13} 1 \\ 9x - 4y \equiv_{13} 10. \end{cases}$$

Jak widać, operujemy działaniami w zbiorze $\mathbb{Z}_{13}$ - pamiętajmy więc, że mamy do dyspozycji tylko liczby całkowite od 0 do 12 i nie możemy używać ułamków!

Układ kongruencji - przykład

Zadanie z egzaminu (2014, zaoczne, I termin)

$$\begin{cases} 3x - 5y \equiv_{13} 1 \\ 9x - 4y \equiv_{13} 10. \end{cases}$$

Jak widać, operujemy działaniami w zbiorze $\mathbb{Z}_{13}$ - pamiętajmy więc, że mamy do dyspozycji tylko liczby całkowite od 0 do 12 i nie możemy używać ułamków!
Jest wiele sposobów rozwiązania tego zadania, ja wybiorę niekoniecznie najprostszy, ale łatwy do uogólnienia i pokazujący wiele poznanych przed chwilą mechanizmów.

Układ kongruencji - przykład

Zadanie z egzaminu (2014, zaoczne, I termin)

$$\begin{cases} 3x - 5y \equiv_{13} 1 \\ 9x - 4y \equiv_{13} 10. \end{cases}$$

Układ kongruencji - przykład

Zadanie z egzaminu (2014, zaoczne, I termin)

$$\begin{cases} 3x - 5y \equiv_{13} 1 \\ 9x - 4y \equiv_{13} 10. \end{cases}$$

Spróbuję w pewnym momencie pozbyć się zmiennej x odejmując kongruencje stronami (operacja dozwolona!). W tym celu chcę drugą kongruencję przekształcić tak, by zamiast $9x$ było w nim $3x$.

Układ kongruencji - przykład

Zadanie z egzaminu (2014, zaoczne, I termin)

$$\begin{cases} 3x - 5y \equiv_{13} 1 \\ 9x - 4y \equiv_{13} 10. \end{cases}$$

Spróbuję w pewnym momencie pozbyć się zmiennej x odejmując kongruencje stronami (operacja dozwolona!). W tym celu chcę drugą kongruencję przekształcić tak, by zamiast $9x$ było w nim $3x$. Najprościej byłoby podzielić obie strony przez 3. Czy to jest dozwolona operacja?

Układ kongruencji - przykład

Zadanie z egzaminu (2014, zaoczne, I termin)

$$\begin{cases} 3x - 5y \equiv_{13} 1 \\ 9x - 4y \equiv_{13} 10. \end{cases}$$

Spróbuję w pewnym momencie pozbyć się zmiennej x odejmując kongruencje stronami (operacja dozwolona!). W tym celu chcę drugą kongruencję przekształcić tak, by zamiast $9x$ było w nim $3x$. Najprościej byłoby podzielić obie strony przez 3. Czy to jest dozwolona operacja? Tak, bo 3 jest względnie pierwsze z 13!

Układ kongruencji - przykład

Zadanie z egzaminu (2014, zaoczne, I termin)

$$\begin{cases} 3x - 5y \equiv_{13} 1 \\ 9x - 4y \equiv_{13} 10. \end{cases}$$

Spróbuję w pewnym momencie pozbyć się zmiennej x odejmując kongruencje stronami (operacja dozwolona!). W tym celu chcę drugą kongruencję przekształcić tak, by zamiast $9x$ było w nim $3x$. Najprościej byłoby podzielić obie strony przez 3. Czy to jest dozwolona operacja? Tak, bo 3 jest względnie pierwsze z 13! Jednak jest problem: liczby $\frac{4}{3}$ i $\frac{10}{3}$ nie istnieją w $\mathbb{Z}_{13}$.

Układ kongruencji - przykład

Zadanie z egzaminu (2014, zaoczne, I termin)

$$\begin{cases} 3x - 5y \equiv_{13} 1 \\ 9x - 4y \equiv_{13} 10. \end{cases}$$

Układ kongruencji - przykład

Zadanie z egzaminu (2014, zaoczne, I termin)

$$\begin{cases} 3x - 5y \equiv_{13} 1 \\ 9x - 4y \equiv_{13} 10. \end{cases}$$

By wykonać dzielenie 4 przez 3 modulo 13, szukam liczby równoważnej 4 w tej arytmetyce, którą potrafię podzielić przez 3.

Układ kongruencji - przykład

Zadanie z egzaminu (2014, zaoczne, I termin)

$$\begin{cases} 3x - 5y \equiv_{13} 1 \\ 9x - 4y \equiv_{13} 10. \end{cases}$$

By wykonać dzielenie 4 przez 3 modulo 13, szukam liczby równoważnej 4 w tej arytmetyce, którą potrafię podzielić przez 3. Będzie to np. 30 (bo $30 = 2 \cdot 13 + 4$), stąd $3 \cdot 10 \equiv_{13} 4$, więc $4 : 3 \equiv_{13} 10$ (ciągle pamiętamy, że $3 \perp 13$, więc dzielenie jest dopuszczalne).

Układ kongruencji - przykład

Zadanie z egzaminu (2014, zaoczne, I termin)

$$\begin{cases} 3x - 5y \equiv_{13} 1 \\ 9x - 4y \equiv_{13} 10. \end{cases}$$

By wykonać dzielenie 4 przez 3 modulo 13, szukam liczby równoważnej 4 w tej arytmetyce, którą potrafię podzielić przez 3. Będzie to np. 30 (bo $30 = 2 \cdot 13 + 4$), stąd $3 \cdot 10 \equiv_{13} 4$, więc $4 : 3 \equiv_{13} 10$ (ciągle pamiętamy, że $3 \perp 13$, więc dzielenie jest dopuszczalne).
By wykonać dzielenie 10 przez 3 modulo 13, szukam liczby równoważnej 10 w tej arytmetyce, którą potrafię podzielić przez 3.

Układ kongruencji - przykład

Zadanie z egzaminu (2014, zaoczne, I termin)

$$\begin{cases} 3x - 5y \equiv_{13} 1 \\ 9x - 4y \equiv_{13} 10. \end{cases}$$

By wykonać dzielenie 4 przez 3 modulo 13, szukam liczby równoważnej 4 w tej arytmetyce, którą potrafię podzielić przez 3. Będzie to np. 30 (bo $30 = 2 \cdot 13 + 4$), stąd $3 \cdot 10 \equiv_{13} 4$, więc $4 : 3 \equiv_{13} 10$ (ciągle pamiętamy, że $3 \perp 13$, więc dzielenie jest dopuszczalne).
By wykonać dzielenie 10 przez 3 modulo 13, szukam liczby równoważnej 10 w tej arytmetyce, którą potrafię podzielić przez 3. Będzie to np. 36 (bo $36 = 2 \cdot 13 + 10$), stąd $3 \cdot 12 \equiv_{13} 10$, więc $10 : 3 \equiv_{13} 12$ (ciągle pamiętamy, że $3 \perp 13$, więc dzielenie jest dopuszczalne).

Układ kongruencji - przykład

Zadanie z egzaminu (2014, zaoczne, I termin)

$$\begin{cases} 3x - 5y \equiv_{13} 1 \\ 9x - 4y \equiv_{13} 10. \end{cases}$$

By wykonać dzielenie 4 przez 3 modulo 13, szukam liczby równoważnej 4 w tej arytmetyce, którą potrafię podzielić przez 3. Będzie to np. 30 (bo $30 = 2 \cdot 13 + 4$), stąd $3 \cdot 10 \equiv_{13} 4$, więc $4 : 3 \equiv_{13} 10$ (ciągle pamiętamy, że $3 \perp 13$, więc dzielenie jest dopuszczalne).
By wykonać dzielenie 10 przez 3 modulo 13, szukam liczby równoważnej 10 w tej arytmetyce, którą potrafię podzielić przez 3. Będzie to np. 36 (bo $36 = 2 \cdot 13 + 10$), stąd $3 \cdot 12 \equiv_{13} 10$, więc $10 : 3 \equiv_{13} 12$ (ciągle pamiętamy, że $3 \perp 13$, więc dzielenie jest dopuszczalne).

Układ kongruencji - przykład

Zadanie z egzaminu (2014, zaoczne, I termin)

$$\begin{cases} 3x - 5y \equiv_{13} 1 \\ 9x - 4y \equiv_{13} 10. \end{cases}$$

Układ kongruencji - przykład

Zadanie z egzaminu (2014, zaoczne, I termin)

$$\begin{cases} 3x - 5y \equiv_{13} 1 \\ 9x - 4y \equiv_{13} 10. \end{cases}$$

Zatem powyższy układ kongruencji mogę zapisać następująco:

$$\begin{cases} 3x - 5y \equiv_{13} 1 \\ 3x - 10y \equiv_{13} 12. \end{cases}$$

Układ kongruencji - przykład

Zadanie

$$\begin{cases} 3x - 5y \equiv_{13} 1 \\ 3x - 10y \equiv_{13} 12. \end{cases}$$

Układ kongruencji - przykład

Zadanie

$$\begin{cases} 3x - 5y \equiv_{13} 1 \\ 3x - 10y \equiv_{13} 12. \end{cases}$$

Teraz odejmę stronami kongruencji (uwzględniając, że $1 - 12 \equiv_{13} 2$, otrzymując

$$5y \equiv_{13} 2.$$

Układ kongruencji - przykład

Zadanie

$$\begin{cases} 3x - 5y \equiv_{13} 1 \\ 3x - 10y \equiv_{13} 12. \end{cases}$$

Teraz odejmę stronami kongruencji (uwzględniając, że $1 - 12 \equiv_{13} 2$, otrzymując

$$5y \equiv_{13} 2.$$

Stąd $y \equiv_{13} 3$ (bo $5 \cdot 3 = 15 \equiv_{13} 2$), a $x \equiv_{13} 1$ (łatwe obliczenie po wstawieniu 3 za y do dowolnej kongruencji).

Układ kongruencji - przykład

Zadanie

$$\begin{cases} 3x - 5y \equiv_{13} 1 \\ 3x - 10y \equiv_{13} 12. \end{cases}$$

Teraz odejmę stronami kongruencji (uwzględniając, że $1 - 12 \equiv_{13} 2$, otrzymując

$$5y \equiv_{13} 2.$$

Stąd $y \equiv_{13} 3$ (bo $5 \cdot 3 = 15 \equiv_{13} 2$), a $x \equiv_{13} 1$ (łatwe obliczenie po wstawieniu 3 za y do dowolnej kongruencji). Warto na koniec sprawdzić obliczenia podstawiając wynik do wyjściowych kongruencji.

Istnienie rozwiązań kongruencji liniowych

Tak jak przy rozwiązywaniu równań liniowych i ich układów, powstaje pytanie, czy kongruencje liniowe i ich układy mają zawsze rozwiązanie i czy to rozwiązanie, jeśli istnieje, jest jedyne.

Istnienie rozwiązań kongruencji liniowych

Tak jak przy rozwiązywaniu równań liniowych i ich układów, powstaje pytanie, czy kongruencje liniowe i ich układy mają zawsze rozwiązanie i czy to rozwiązanie, jeśli istnieje, jest jedyne. I również teraz odpowiedź jest negatywna: kongruencje liniowe i ich układy, tak jak równania i ich układy, mogą nie mieć rozwiązań, mogą mieć jedno rozwiązanie lub (i tu jest różnica w stosunku do układów równań liniowych) mieć ich kilka, ale nie nieskończenie wiele.

Istnienie rozwiązań kongruencji liniowych

Tak jak przy rozwiązywaniu równań liniowych i ich układów, powstaje pytanie, czy kongruencje liniowe i ich układy mają zawsze rozwiązanie i czy to rozwiązanie, jeśli istnieje, jest jedyne. I również teraz odpowiedź jest negatywna: kongruencje liniowe i ich układy, tak jak równania i ich układy, mogą nie mieć rozwiązań, mogą mieć jedno rozwiązanie lub (i tu jest różnica w stosunku do układów równań liniowych) mieć ich kilka, ale nie nieskończenie wiele.
Przykładowo, kongruencja $2x \equiv_4 3$ nie posiada rozwiązań, zaś kongruencja $2x \equiv_4 2$ ma dwa rozwiązania.

Istnienie rozwiązań układów kongruencji

Podobnie układ kongruencji:

$$\begin{cases} 3x + y \equiv_{17} 1 \\ x + 6y \equiv_{17} 2 \end{cases}$$

nie ma rozwiązań,

Istnienie rozwiązań układów kongruencji

Podobnie układ kongruencji:

$$\begin{cases} 3x + y \equiv_{17} 1 \\ x + 6y \equiv_{17} 2 \end{cases}$$

nie ma rozwiązań, a układ:

$$\begin{cases} 3x + y \equiv_{17} 1 \\ x + 6y \equiv_{17} 6 \end{cases}$$

ma aż 17 rozwiązań.

Istnienie rozwiązań układów kongruencji

Dla układów równań liniowych, kwestię istnienia i jednoznaczności rozwiązań rozstrzyga twierdzenie Kroneckera-Capellego.

Istnienie rozwiązań układów kongruencji

Dla układów równań liniowych, kwestię istnienia i jednoznaczności rozwiązań rozstrzyga twierdzenie Kroneckera-Capellego. W wypadku kongruencji, odpowiednik byłby nieco bardziej skomplikowany. Dlatego zadowolimy się poniższymi wynikami częściowymi:

Istnienie rozwiązań układów kongruencji

Dla układów równań liniowych, kwestię istnienia i jednoznaczności rozwiązań rozstrzyga twierdzenie Kroneckera-Capellego. W wypadku kongruencji, odpowiednik byłby nieco bardziej skomplikowany. Dlatego zadowolimy się poniższymi wynikami częściowymi:

Rozwiązalność kongruencji

Kongruencja liniowa $ax \equiv_n b$ ma co najmniej jedno rozwiązanie wtedy i tylko wtedy gdy $NWD(a, n)|b$.

Istnienie rozwiązań układów kongruencji

Dla układów równań liniowych, kwestię istnienia i jednoznaczności rozwiązań rozstrzyga twierdzenie Kroneckera-Capellego. W wypadku kongruencji, odpowiednik byłby nieco bardziej skomplikowany. Dlatego zadowolimy się poniższymi wynikami częściowymi:

Rozwiązalność kongruencji

Kongruencja liniowa $ax \equiv_n b$ ma co najmniej jedno rozwiązanie wtedy i tylko wtedy gdy $NWD(a, n)|b$.

Rozwiązalność układów kongruencji

Jeśli $\det A \perp n$ to układ kongruencji liniowych $Ax \equiv_n b$ ma dokładnie jedno rozwiązanie.

Istnienie rozwiązań układów kongruencji

Dla układów równań liniowych, kwestię istnienia i jednoznaczności rozwiązań rozstrzyga twierdzenie Kroneckera-Capellego. W wypadku kongruencji, odpowiednik byłby nieco bardziej skomplikowany. Dlatego zadowolimy się poniższymi wynikami częściowymi:

Rozwiązalność kongruencji

Kongruencja liniowa $ax \equiv_n b$ ma co najmniej jedno rozwiązanie wtedy i tylko wtedy gdy $NWD(a, n)|b$.

Rozwiązalność układów kongruencji

Jeśli $\det A \perp n$ to układ kongruencji liniowych $Ax \equiv_n b$ ma dokładnie jedno rozwiązanie.

Arytmetyka modularna - zastosowania

Pierwsze zastosowanie dotyczy bardziej pojęcia podzielności, niż arytmetyki modularnej, ale arytmetyka modularna może znacząco pomóc w ulepszeniu tego zastosowania.

Arytmetyka modularna - zastosowania

Pierwsze zastosowanie dotyczy bardziej pojęcia podzielności, niż arytmetyki modularnej, ale arytmetyka modularna może znacząco pomóc w ulepszeniu tego zastosowania.
Zagadnienie dotyczy autokorygujących kodów stosowanych w wielu zagadnieniach, w których łatwo o błędy w odtwarzaniu kodu (numer karty kredytowej, konta bankowego, zapis cyfrowy muzyki itp.).

Arytmetyka modularna - zastosowania

Pierwsze zastosowanie dotyczy bardziej pojęcia podzielności, niż arytmetyki modularnej, ale arytmetyka modularna może znacząco pomóc w ulepszeniu tego zastosowania.
Zagadnienie dotyczy autokorygujących kodów stosowanych w wielu zagadnieniach, w których łatwo o błędy w odtwarzaniu kodu (numer karty kredytowej, konta bankowego, zapis cyfrowy muzyki itp.).
Powiedzmy, że chcemy przydzielić numery kont kilkudziesięciu tysiącom ludzi. Teoretycznie wystarczy im przypisać numery pięciocyfrowe.

Arytmetyka modularna - zastosowania

Pierwsze zastosowanie dotyczy bardziej pojęcia podzielności, niż arytmetyki modularnej, ale arytmetyka modularna może znacząco pomóc w ulepszeniu tego zastosowania.
Zagadnienie dotyczy autokorygujących kodów stosowanych w wielu zagadnieniach, w których łatwo o błędy w odtwarzaniu kodu (numer karty kredytowej, konta bankowego, zapis cyfrowy muzyki itp.).
Powiedzmy, że chcemy przydzielić numery kont kilkudziesięciu tysiącom ludzi. Teoretycznie wystarczy im przypisać numery pięciocyfrowe. Jednakże, próbując wysłać przelew na czyjeś konto, łatwo pomylić jakąś cyfrę i przesłać pieniądze komuś zupełnie innemu.

Arytmetyka modularna - zastosowania

Pierwsze zastosowanie dotyczy bardziej pojęcia podzielności, niż arytmetyki modularnej, ale arytmetyka modularna może znacząco pomóc w ulepszeniu tego zastosowania.
Zagadnienie dotyczy autokorygujących kodów stosowanych w wielu zagadnieniach, w których łatwo o błędy w odtwarzaniu kodu (numer karty kredytowej, konta bankowego, zapis cyfrowy muzyki itp.).
Powiedzmy, że chcemy przydzielić numery kont kilkudziesięciu tysiącom ludzi. Teoretycznie wystarczy im przypisać numery pięciocyfrowe. Jednakże, próbując wysłać przelew na czyjeś konto, łatwo pomylić jakąś cyfrę i przesłać pieniądze komuś zupełnie innemu. Naprawianie takich pomyłek jest dość kosztowne i czasochłonne, dlatego lepiej je uniemożliwić.

Arytmetyka modularna - zastosowania

Stąd pomysł tzw. „cyfr kontrolnych”, które obecnie pojawiają się w każdym kodzie tego typu. Niektóre cyfry nie są numerem danego obiektu lecz pełnią funkcję badania poprawności całego kodu.

Arytmetyka modularna - zastosowania

Stąd pomysł tzw. „cyfr kontrolnych", które obecnie pojawiają się w każdym kodzie tego typu. Niektóre cyfry nie są numerem danego obiektu lecz pełnią funkcję badania poprawności całego kodu. Najprostszym przykładem w naszej sytuacji byłoby dodanie do kodu szóstej cyfry, która powstaje w następujący sposób:

Arytmetyka modularna - zastosowania

Stąd pomysł tzw. „cyfr kontrolnych", które obecnie pojawiają się w każdym kodzie tego typu. Niektóre cyfry nie są numerem danego obiektu lecz pełnią funkcję badania poprawności całego kodu. Najprostszym przykładem w naszej sytuacji byłoby dodanie do kodu szóstej cyfry, która powstaje w następujący sposób: dodajemy do siebie pięć pozostałych cyfr i reszta z dzielenia tej sumy przez 10 będzie szóstą cyfrą.

Arytmetyka modularna - zastosowania

Stąd pomysł tzw. „cyfr kontrolnych", które obecnie pojawiają się w każdym kodzie tego typu. Niektóre cyfry nie są numerem danego obiektu lecz pełnią funkcję badania poprawności całego kodu. Najprostszym przykładem w naszej sytuacji byłoby dodanie do kodu szóstej cyfry, która powstaje w następujący sposób: dodajemy do siebie pięć pozostałych cyfr i reszta z dzielenia tej sumy przez 10 będzie szóstą cyfrą. Na przykład klient o numerze 39605 miałby sumę cyfr równą 23,

Arytmetyka modularna - zastosowania

Stąd pomysł tzw. „cyfr kontrolnych", które obecnie pojawiają się w każdym kodzie tego typu. Niektóre cyfry nie są numerem danego obiektu lecz pełnią funkcję badania poprawności całego kodu. Najprostszym przykładem w naszej sytuacji byłoby dodanie do kodu szóstej cyfry, która powstaje w następujący sposób: dodajemy do siebie pięć pozostałych cyfr i reszta z dzielenia tej sumy przez 10 będzie szóstą cyfrą. Na przykład klient o numerze 39605 miałby sumę cyfr równą 23, więc po zbadaniu reszty z dzielenia przez 10 ostateczny numer jego konta to:

$$396053.$$

Arytmetyka modularna - zastosowania

Stąd pomysł tzw. „cyfr kontrolnych", które obecnie pojawiają się w każdym kodzie tego typu. Niektóre cyfry nie są numerem danego obiektu lecz pełnią funkcję badania poprawności całego kodu. Najprostszym przykładem w naszej sytuacji byłoby dodanie do kodu szóstej cyfry, która powstaje w następujący sposób: dodajemy do siebie pięć pozostałych cyfr i reszta z dzielenia tej sumy przez 10 będzie szóstą cyfrą. Na przykład klient o numerze 39605 miałby sumę cyfr równą 23, więc po zbadaniu reszty z dzielenia przez 10 ostateczny numer jego konta to:

$$396053.$$

Takie przypisanie chroni przed błędnym wpisaniem jednej cyfry kodu. Np. jeśli ktoś wziąłby niewyraźnie napisane 0 za kolejną cyfrę 6 i próbował wysłać przelew na konto numer 396653, natychmiast otrzymałby odpowiedź, że taki numer konta jest niemożliwy, gdyż suma cyfr $3, 9, 6, 6, 5$ daje resztę 9, a nie 3.

Arytmetyka modularna - zastosowania

Oczywiście, przedstawiony mechanizm jest bardzo uproszczony w stosunku do stosowanych w praktyce

Arytmetyka modularna - zastosowania

Oczywiście, przedstawiony mechanizm jest bardzo uproszczony w stosunku do stosowanych w praktyce (nie chroni np. przed „czeskim błędem", czyli zapisaniem dwu cyfr w zamienionej kolejności) i w nich arytmetyka modularna i jej prawa przydają się bardziej.

Arytmetyka modularna - zastosowania

Oczywiście, przedstawiony mechanizm jest bardzo uproszczony w stosunku do stosowanych w praktyce (nie chroni np. przed „czeskim błędem", czyli zapisaniem dwu cyfr w zamienionej kolejności) i w nich arytmetyka modularna i jej prawa przydają się bardziej. Zachęcam do prób udoskonalania tego typu autokorygujących kodów na ćwiczeniach lub samodzielnie.

Arytmetyka modularna - integer overflow

Naturalny sposób przechowywania liczb w pamięci komputera oznacza, że zmienne pozornie całkowitoliczbowe pochodzą tak naprawdę ze zbiorów $\mathbb{Z}_n$. Jest to źródłem błędów programistycznych typu (*integer overflow*).
Jeśli w takiej sytuacji zmienna takiego typu zwiększa swoją wartość (np. indeks tablicy), to w pewnym momencie może dojść do *przekręcenia licznika*, który z wartości maksymalnych przeskoczy nagle na minimalne.
Przykłady: lot rakiety Ariane 5 (1996), legenda o "atomowym Gandhim".

Modularny wiedźmin

Z innym „praktycznym” zadaniem z arytmetyki modularnej natknąłem się w grze *Wiedźmin 2: Zabójcy królów*.

Modularny wiedźmin

Z innym „praktycznym” zadaniem z arytmetyki modularnej natknąłem się w grze *Wiedźmin 2: Zabójcy królów*.
Gracz miał tam do rozwiązania następującą zagadkę: w sali było 7 (zapewne magicznych) palenisk. Na początku wszystkie były wygaszone, należało wszystkie rozpalić.

Modularny wiedźmin

Z innym „praktycznym” zadaniem z arytmetyki modularnej natknąłem się w grze *Wiedźmin 2: Zabójcy królów*.
Gracz miał tam do rozwiązania następującą zagadkę: w sali było 7 (zapewne magicznych) palenisk. Na początku wszystkie były wygaszone, należało wszystkie rozpalić. Problem w tym, że zmiana stanu jednego z palenisk powodowała jednocześnie zmianę stanu dwóch innych.

Modularny wiedźmin

Z innym „praktycznym” zadaniem z arytmetyki modularnej natknąłem się w grze *Wiedźmin 2: Zabójcy królów*.
Gracz miał tam do rozwiązania następującą zagadkę: w sali było 7 (zapewne magicznych) palenisk. Na początku wszystkie były wygaszone, należało wszystkie rozpalić. Problem w tym, że zmiana stanu jednego z palenisk powodowała jednocześnie zmianę stanu dwóch innych. I tak: próba rozpalenia (lub zgaszenia) pierwszego paleniska rozpalała (lub gasiła) paleniska nr 4 i 6,

Modularny wiedźmin

Z innym „praktycznym” zadaniem z arytmetyki modularnej natknąłem się w grze *Wiedźmin 2: Zabójcy królów*.
Gracz miał tam do rozwiązania następującą zagadkę: w sali było 7 (zapewne magicznych) palenisk. Na początku wszystkie były wygaszone, należało wszystkie rozpalić. Problem w tym, że zmiana stanu jednego z palenisk powodowała jednocześnie zmianę stanu dwóch innych. I tak: próba rozpalenia (lub zgaszenia) pierwszego paleniska rozpalała (lub gasiła) paleniska nr 4 i 6, rozpalanie/gaszenie drugiego paleniska zmieniało stan palenisk 1 i 7, rozpalanie/gaszenie trzeciego paleniska zmieniało stan palenisk 2 i 5, rozpalanie/gaszenie czwartego paleniska zmieniało stan palenisk 6 i 7, rozpalanie/gaszenie piątego paleniska zmieniało stan palenisk 2 i 4, rozpalanie/gaszenie szóstego paleniska zmieniało stan palenisk 2 i 5 i wreszcie rozpalanie/gaszenie siódmego paleniska zmieniało stan palenisk 1 i 3.

Modularny wiedźmin

Rozwiązując ten problem na chybił-trafił ma się do sprawdzenia 128 przypadków. Nie tak dużo, ale trochę czasu zajmuje.

Modularny wiedźmin

Rozwiązując ten problem na chybił-trafił ma się do sprawdzenia 128 przypadków. Nie tak dużo, ale trochę czasu zajmuje. Tymczasem naturalnym sposobem rozwiązania tego zagadnienia jest arytmetyka modularna.

Modularny wiedźmin

Rozwiązując ten problem na chybił-trafił ma się do sprawdzenia 128 przypadków. Nie tak dużo, ale trochę czasu zajmuje. Tymczasem naturalnym sposobem rozwiązania tego zagadnienia jest arytmetyka modularna. Wystarczy zauważyć, że tak naprawdę nie jest ważne ile razy dokładnie dane palenisko zmieni swój stan, ale czy uczyni to parzystą, czy nieparzystą ilość razy (bo dwukrotna zmiana stanu powoduje powrót do stanu pierwotnego).

Modularny wiedźmin

Rozwiązując ten problem na chybił-trafił ma się do sprawdzenia 128 przypadków. Nie tak dużo, ale trochę czasu zajmuje. Tymczasem naturalnym sposobem rozwiązania tego zagadnienia jest arytmetyka modularna. Wystarczy zauważyć, że tak naprawdę nie jest ważne ile razy dokładnie dane palenisko zmieni swój stan, ale czy uczyni to parzystą, czy nieparzystą ilość razy (bo dwukrotna zmiana stanu powoduje powrót do stanu pierwotnego). A parzystość i nieparzystość liczby jest determinowana przez jej wartość modulo 2.

Modularny wiedźmin

Rozwiązując ten problem na chybił-trafił ma się do sprawdzenia 128 przypadków. Nie tak dużo, ale trochę czasu zajmuje. Tymczasem naturalnym sposobem rozwiązania tego zagadnienia jest arytmetyka modularna. Wystarczy zauważyć, że tak naprawdę nie jest ważne ile razy dokładnie dane palenisko zmieni swój stan, ale czy uczyni to parzystą, czy nieparzystą ilość razy (bo dwukrotna zmiana stanu powoduje powrót do stanu pierwotnego). A parzystość i nieparzystość liczby jest determinowana przez jej wartość modulo 2. Dlatego, jeśli przez x_1, x_2, ... x_7 oznaczymy liczbę koniecznych rozpaleń palenisk $1, 2, \ldots, 7$, to wystarczy rozwiązać łatwy układ 7 równań modulo 2 z 7 niewiadomymi.

Modularny wiedźmin

Rozwiązując ten problem na chybił-trafił ma się do sprawdzenia 128 przypadków. Nie tak dużo, ale trochę czasu zajmuje. Tymczasem naturalnym sposobem rozwiązania tego zagadnienia jest arytmetyka modularna. Wystarczy zauważyć, że tak naprawdę nie jest ważne ile razy dokładnie dane palenisko zmieni swój stan, ale czy uczyni to parzystą, czy nieparzystą ilość razy (bo dwukrotna zmiana stanu powoduje powrót do stanu pierwotnego). A parzystość i nieparzystość liczby jest determinowana przez jej wartość modulo 2. Dlatego, jeśli przez x_1, x_2, ... x_7 oznaczymy liczbę koniecznych rozpaleń palenisk $1, 2, \ldots, 7$, to wystarczy rozwiązać łatwy układ 7 równań modulo 2 z 7 niewiadomymi. Na przykład pierwsze palenisko rozpalamy za pomocą operacji x_1, x_2 i x_7 i musimy uczynić to nieparzystą liczbę razy, co zapiszemy tak:

$$x_1 + x_2 + x_7 \equiv_2 1.$$

Modularny wiedźmin

Pełny układ równań to:

$$\begin{cases} x_1 + x_2 + x_7 \equiv_2 1 \\ x_2 + x_3 + x_5 + x_6 \equiv_2 1 \\ x_3 + x_7 \equiv_2 1 \\ x_1 + x_4 + x_5 \equiv_2 1 \\ x_3 + x_5 + x_6 \equiv_2 1 \\ x_2 + x_4 + x_7 \equiv_2 1 \end{cases}$$

Co ciekawe, układ ten ma dokładnie 2 rozwiązania - pozostawiam to Państwu na zadanie domowe.

Funkcja Eulera - wstęp

Zanim przejdziemy do kolejnego działu, by przedstawić zastosowania teorii liczb we współczesnej kryptografii, potrzebujemy jeszcze jednego pojęcia i kilku twierdzeń z nim związanych.

Funkcja Eulera - wstęp

Zanim przejdziemy do kolejnego działu, by przedstawić zastosowania teorii liczb we współczesnej kryptografii, potrzebujemy jeszcze jednego pojęcia i kilku twierdzeń z nim związanych. Chodzi o tzw. funkcję φ Eulera (to nie ostatni raz, kiedy to nazwisko się w tym kursie pojawia!).

Funkcja Eulera - definicja

Funkcja φ Eulera

Funkcja φ Eulera (zwana tocjentem) to $\varphi : \mathbb{N} \setminus \{0\} \longrightarrow \mathbb{N}$ zdefiniowana wzorem:

$$\varphi(n) = \left|\{1 \leqslant a \leqslant n \quad : \quad \mathrm{NWD}(a, n) = 1\}\right|,$$

czyli jest to odwzorowanie przyporządkowujące liczbie n moc zbioru liczb naturalnych dodatnich nie większych od niej i względnie pierwszych z nią.

Funkcja Eulera - definicja

Funkcja φ Eulera

Funkcja φ Eulera (zwana tocjentem) to $\varphi : \mathbb{N} \setminus \{0\} \longrightarrow \mathbb{N}$ zdefiniowana wzorem:

$$\varphi(n) = \left|\{1 \leqslant a \leqslant n \quad : \quad \mathrm{NWD}(a, n) = 1\}\right|,$$

czyli jest to odwzorowanie przyporządkowujące liczbie n moc zbioru liczb naturalnych dodatnich nie większych od niej i względnie pierwszych z nią.

Przykładowo obliczmy $\varphi(6)$.

Funkcja Eulera - definicja

Funkcja φ Eulera

Funkcja φ Eulera (zwana tocjentem) to $\varphi : \mathbb{N} \setminus \{0\} \longrightarrow \mathbb{N}$ zdefiniowana wzorem:

$$\varphi(n) = \left|\{1 \leqslant a \leqslant n \quad : \quad \mathrm{NWD}(a, n) = 1\}\right|,$$

czyli jest to odwzorowanie przyporządkowujące liczbie n moc zbioru liczb naturalnych dodatnich nie większych od niej i względnie pierwszych z nią.

Przykładowo obliczmy $\varphi(6)$. Liczby nie większe od niej i względnie pierwsze z nią to

Funkcja Eulera - definicja

Funkcja φ Eulera

Funkcja φ Eulera (zwana tocjentem) to $\varphi : \mathbb{N} \setminus \{0\} \longrightarrow \mathbb{N}$ zdefiniowana wzorem:

$$\varphi(n) = \left|\{1 \leqslant a \leqslant n \quad : \quad \mathrm{NWD}(a, n) = 1\}\right|,$$

czyli jest to odwzorowanie przyporządkowujące liczbie n moc zbioru liczb naturalnych dodatnich nie większych od niej i względnie pierwszych z nią.

Przykładowo obliczmy $\varphi(6)$. Liczby nie większe od niej i względnie pierwsze z nią to 1 i 5, więc $\varphi(6) =$

Funkcja Eulera - definicja

Funkcja φ Eulera

Funkcja φ Eulera (zwana tocjentem) to $\varphi : \mathbb{N} \setminus \{0\} \longrightarrow \mathbb{N}$ zdefiniowana wzorem:

$$\varphi(n) = \left|\{1 \leqslant a \leqslant n \quad : \quad \mathrm{NWD}(a, n) = 1\}\right|,$$

czyli jest to odwzorowanie przyporządkowujące liczbie n moc zbioru liczb naturalnych dodatnich nie większych od niej i względnie pierwszych z nią.

Przykładowo obliczmy $\varphi(6)$. Liczby nie większe od niej i względnie pierwsze z nią to 1 i 5, więc $\varphi(6) = 2$.

Funkcja Eulera - definicja

Funkcja φ Eulera

Funkcja φ Eulera (zwana tocjentem) to $\varphi : \mathbb{N} \setminus \{0\} \longrightarrow \mathbb{N}$ zdefiniowana wzorem:

$$\varphi(n) = \left|\{1 \leqslant a \leqslant n \quad : \quad \mathrm{NWD}(a, n) = 1\}\right|,$$

czyli jest to odwzorowanie przyporządkowujące liczbie n moc zbioru liczb naturalnych dodatnich nie większych od niej i względnie pierwszych z nią.

Przykładowo obliczmy $\varphi(6)$. Liczby nie większe od niej i względnie pierwsze z nią to 1 i 5, więc $\varphi(6) = 2$.
Oczywiście, nie jest to zbyt wygodny sposób obliczania wartości funkcji φ.

Funkcja Eulera - twierdzenia

Funkcja φ Eulera dla liczb pierwszych

Dla dowolnej liczby pierwszej p zachodzą związki:

a) $\varphi(p) = p - 1$

b) $\varphi(p^k) = p^k(1 - \frac{1}{p})$.

Funkcja Eulera - twierdzenia

Funkcja φ Eulera dla liczb pierwszych

Dla dowolnej liczby pierwszej p zachodzą związki:
a) $\varphi(p) = p - 1$
b) $\varphi(p^k) = p^k(1 - \frac{1}{p})$.

Funkcja φ Eulera dla iloczynów liczb względnie pierwszych

Dla dowolnych dwóch dodatnich liczb względnie pierwszych m i n zachodzi:
$\varphi(mn) = \varphi(m)\varphi(n)$.

Funkcja Eulera - twierdzenia

Funkcja φ Eulera dla liczb pierwszych

Dla dowolnej liczby pierwszej p zachodzą związki:
a) $\varphi(p) = p - 1$
b) $\varphi(p^k) = p^k(1 - \frac{1}{p})$.

Funkcja φ Eulera dla iloczynów liczb względnie pierwszych

Dla dowolnych dwóch dodatnich liczb względnie pierwszych m i n zachodzi:
$\varphi(mn) = \varphi(m)\varphi(n)$.

Wniosek - potrafimy policzyć wartość funkcji Eulera dla każdej liczby, której rozkład na czynniki pierwsze znamy.

Funkcja Eulera - przykład

Zadanie

Obliczyć $\varphi(600)$

Funkcja Eulera - przykład

Zadanie

Obliczyć $\varphi(600)$

Łatwo zauważyć, że $600 = 2^3 \cdot 3 \cdot 5^2$.

Funkcja Eulera - przykład

Zadanie

Obliczyć $\varphi(600)$

Łatwo zauważyć, że $600 = 2^3 \cdot 3 \cdot 5^2$. Dodatkowo, oczywiście 2^3, 3 i 5^2 są parami względnie pierwsze, więc możemy skorzystać z obu twierdzeń z poprzedniego slajdu, otrzymując:

$$\varphi(600) = \varphi(2^3 \cdot 3 \cdot 5^2)$$

Funkcja Eulera - przykład

Zadanie

Obliczyć $\varphi(600)$

Łatwo zauważyć, że $600 = 2^3 \cdot 3 \cdot 5^2$. Dodatkowo, oczywiście 2^3, 3 i 5^2 są parami względnie pierwsze, więc możemy skorzystać z obu twierdzeń z poprzedniego slajdu, otrzymując:

$$\varphi(600) = \varphi(2^3 \cdot 3 \cdot 5^2) = \varphi(2^3) \cdot \varphi(3) \cdot \varphi(5^2)$$

Funkcja Eulera - przykład

Zadanie

Obliczyć $\varphi(600)$

Łatwo zauważyć, że $600 = 2^3 \cdot 3 \cdot 5^2$. Dodatkowo, oczywiście 2^3, 3 i 5^2 są parami względnie pierwsze, więc możemy skorzystać z obu twierdzeń z poprzedniego slajdu, otrzymując:

$$\varphi(600) = \varphi(2^3 \cdot 3 \cdot 5^2) = \varphi(2^3) \cdot \varphi(3) \cdot \varphi(5^2) = 2^3(1-\frac{1}{2}) \cdot 2 \cdot 5^2(1-\frac{1}{5})$$

Funkcja Eulera - przykład

Zadanie

Obliczyć $\varphi(600)$

Łatwo zauważyć, że $600 = 2^3 \cdot 3 \cdot 5^2$. Dodatkowo, oczywiście 2^3, 3 i 5^2 są parami względnie pierwsze, więc możemy skorzystać z obu twierdzeń z poprzedniego slajdu, otrzymując:

$$\varphi(600) = \varphi(2^3 \cdot 3 \cdot 5^2) = \varphi(2^3) \cdot \varphi(3) \cdot \varphi(5^2) = 2^3(1-\frac{1}{2}) \cdot 2 \cdot 5^2(1-\frac{1}{5}) = 160.$$

Obliczanie funkcji Eulera - ważne uwagi

Wydaje się, że z łatwością potrafimy w takim razie obliczyć funkcję Eulera dla każdej liczby.

Obliczanie funkcji Eulera - ważne uwagi

Wydaje się, że z łatwością potrafimy w takim razie obliczyć funkcję Eulera dla każdej liczby. Jednak, trzeba tu podkreślić, że to jest faktycznie łatwe tylko dla liczb, które potrafimy rozłożyć na czynniki pierwsze.

Obliczanie funkcji Eulera - ważne uwagi

Wydaje się, że z łatwością potrafimy w takim razie obliczyć funkcję Eulera dla każdej liczby. Jednak, trzeba tu podkreślić, że to jest faktycznie łatwe tylko dla liczb, które potrafimy rozłożyć na czynniki pierwsze. A to jest niezwykle trudne, nawet dla komputerów, jeśli mówimy o dużych liczbach.

Obliczanie funkcji Eulera - ważne uwagi

Wydaje się, że z łatwością potrafimy w takim razie obliczyć funkcję Eulera dla każdej liczby. Jednak, trzeba tu podkreślić, że to jest faktycznie łatwe tylko dla liczb, które potrafimy rozłożyć na czynniki pierwsze. A to jest niezwykle trudne, nawet dla komputerów, jeśli mówimy o dużych liczbach.
Niestety (a może na szczęście, dla bezpieczeństwa różnych systemów) nie ma prostszego sposobu obliczania funkcji Eulera niż rozkład liczby na czynniki pierwsze i skorzystanie z twierdzeń. Dlatego można powiedzieć, że obliczenie wartości funkcji Eulera jest równie trudne jak rozkład liczb na czynniki pierwsze.

Twierdzenie Eulera i Małe Twierdzenie Fermata

Twierdzenie Eulera

Jeśli $a \perp n$ to $a^{\varphi(n)} \equiv_n 1$.

Twierdzenie Eulera i Małe Twierdzenie Fermata

Twierdzenie Eulera

Jeśli $a \perp n$ to $a^{\varphi(n)} \equiv_n 1$.

Szczególnie użytecznym wnioskiem z twierdzenia Eulera jest:

Małe Twierdzenie Fermata

Dla dowolnej liczby pierwszej p i dodatniego n zachodzi $n^p \equiv_p n$.

Twierdzenie Eulera i Małe Twierdzenie Fermata

Twierdzenie Eulera

Jeśli $a \perp n$ to $a^{\varphi(n)} \equiv_n 1$.

Szczególnie użytecznym wnioskiem z twierdzenia Eulera jest:

Małe Twierdzenie Fermata

Dla dowolnej liczby pierwszej p i dodatniego n zachodzi $n^p \equiv_p n$.

Obydwa te twierdzenia są szalenie użyteczne w kryptografii, gdyż ułatwiają arytmetykę modularną na dużych liczbach.

Twierdzenie Eulera - przykład

Zadanie

Obliczyć $19^{74} \mod 28$

Twierdzenie Eulera - przykład

Zadanie

Obliczyć $19^{74} \mod 28$

19^{74} jest liczbą gigantyczną i liczenie tego „na rympał” bez pomocy sztuczek (nawet przy pomocy komputera) jest koszmarne.

Twierdzenie Eulera - przykład

Zadanie

Obliczyć $19^{74} \mod 28$

19^{74} jest liczbą gigantyczną i liczenie tego „na rympał” bez pomocy sztuczek (nawet przy pomocy komputera) jest koszmarne.

Twierdzenie Eulera

Jeśli $a \perp n$ to $a^{\varphi(n)} \equiv_n 1$.

Twierdzenie Eulera - przykład

Zadanie

Obliczyć $19^{74} \mod 28$

19^{74} jest liczbą gigantyczną i liczenie tego „na rympał” bez pomocy sztuczek (nawet przy pomocy komputera) jest koszmarne.

Twierdzenie Eulera

Jeśli $a \perp n$ to $a^{\varphi(n)} \equiv_n 1$.

Oczywiście $19 \perp 28$ (bo 19 jest liczbą pierwszą), więc możemy skorzystać z twierdzenia Eulera.

Twierdzenie Eulera - przykład

Zadanie

Obliczyć $19^{74} \mod 28$

19^{74} jest liczbą gigantyczną i liczenie tego „na rympał” bez pomocy sztuczek (nawet przy pomocy komputera) jest koszmarne.

Twierdzenie Eulera

Jeśli $a \perp n$ to $a^{\varphi(n)} \equiv_n 1$.

Oczywiście $19 \perp 28$ (bo 19 jest liczbą pierwszą), więc możemy skorzystać z twierdzenia Eulera. $\varphi(28) = \varphi(2^2 \cdot 7) = 2 \cdot 6 = 12$.

Twierdzenie Eulera - przykład

Zadanie

Obliczyć $19^{74} \mod 28$

19^{74} jest liczbą gigantyczną i liczenie tego „na rympał” bez pomocy sztuczek (nawet przy pomocy komputera) jest koszmarne.

Twierdzenie Eulera

Jeśli $a \perp n$ to $a^{\varphi(n)} \equiv_n 1$.

Oczywiście $19 \perp 28$ (bo 19 jest liczbą pierwszą), więc możemy skorzystać z twierdzenia Eulera. $\varphi(28) = \varphi(2^2 \cdot 7) = 2 \cdot 6 = 12$. Stąd wiemy, że $19^{12} \equiv_{28} 1$.

Twierdzenie Eulera - przykład

Zadanie

Obliczyć $19^{74} \mod 28$

19^{74} jest liczbą gigantyczną i liczenie tego „na rympał" bez pomocy sztuczek (nawet przy pomocy komputera) jest koszmarne.

Twierdzenie Eulera

Jeśli $a \perp n$ to $a^{\varphi(n)} \equiv_n 1$.

Oczywiście $19 \perp 28$ (bo 19 jest liczbą pierwszą), więc możemy skorzystać z twierdzenia Eulera. $\varphi(28) = \varphi(2^2 \cdot 7) = 2 \cdot 6 = 12$. Stąd wiemy, że $19^{12} \equiv_{28} 1$.
$19^{74} = (19^{12})^6 \cdot 19^2$, czyli
$19^{74} \equiv_{28} (19^{12})^6 \cdot 19^2$

Twierdzenie Eulera - przykład

Zadanie

Obliczyć $19^{74} \mod 28$

19^{74} jest liczbą gigantyczną i liczenie tego „na rympał" bez pomocy sztuczek (nawet przy pomocy komputera) jest koszmarne.

Twierdzenie Eulera

Jeśli $a \perp n$ to $a^{\varphi(n)} \equiv_n 1$.

Oczywiście $19 \perp 28$ (bo 19 jest liczbą pierwszą), więc możemy skorzystać z twierdzenia Eulera. $\varphi(28) = \varphi(2^2 \cdot 7) = 2 \cdot 6 = 12$. Stąd wiemy, że $19^{12} \equiv_{28} 1$.
$19^{74} = (19^{12})^6 \cdot 19^2$, czyli
$19^{74} \equiv_{28} (19^{12})^6 \cdot 19^2 \equiv_{28} (1)^6 \cdot 361$

Twierdzenie Eulera - przykład

Zadanie

Obliczyć $19^{74} \mod 28$

19^{74} jest liczbą gigantyczną i liczenie tego „na rympał" bez pomocy sztuczek (nawet przy pomocy komputera) jest koszmarne.

Twierdzenie Eulera

Jeśli $a \perp n$ to $a^{\varphi(n)} \equiv_n 1$.

Oczywiście $19 \perp 28$ (bo 19 jest liczbą pierwszą), więc możemy skorzystać z twierdzenia Eulera. $\varphi(28) = \varphi(2^2 \cdot 7) = 2 \cdot 6 = 12$. Stąd wiemy, że $19^{12} \equiv_{28} 1$.
$19^{74} = (19^{12})^6 \cdot 19^2$, czyli
$19^{74} \equiv_{28} (19^{12})^6 \cdot 19^2 \equiv_{28} (1)^6 \cdot 361 \equiv_{28} 361 \equiv_{28} 25$.