Algebra - čtvrtý díl

Lenka Zalabová

Ústav matematiky a biomatematiky, Přírodovědecká fakulta, Jihočeská univerzita v Českých Budějovicích

zima 2012

Obsah

1. Prvočísla
2. Kongruenční rovnice
3. Testování prvočíselnosti a faktorizace
4. Šifrovací algoritmus RSA

1 Prvočísla

2 Kongruenční rovnice

3 Testování prvočíselnosti a faktorizace

4 Šifrovací algoritmus RSA

Prvočísla

Definition

Přirozené číslo $p > 1$ se nazývá *prvočíslo*, jestliže jeho jediným dělitelem větším než 1 je p samotné.

1. Libovolné přirozené číslo $n > 1$ je buď prvočíslo, nebo jej lze právě jedním způsobem rozložit na součin prvočísel.
2. Prvočísel je nekonečně mnoho.

Eratosthenovo síto

Jednoduchý algoritmus pro nalezení všech prvočísel menších než zadaná horní mez n.

- Napíšeme všechna čísla od 2 do n.
- První neoznačené číslo v posloupnosti je prvočíslo. Toto číslo si označíme. Z posloupnosti vyškrtáme všechny jeho násobky.
- Takto postupujeme, dokud je v posloupnosti nějaké neoznačené číslo.

Stačí postupovat pouze do doby, dokud není jako prvočíslo označeno číslo vyšší než $\sqrt{n}$. V takové chvíli už všechna zbývající čísla jsou nutně prvočísla.

Eratosthenovo síto

Jednoduchý algoritmus pro nalezení všech prvočísel menších než zadaná horní mez n.

- Napíšeme všechna čísla od 2 do n.
- První neoznačené číslo v posloupnosti je prvočíslo. Toto číslo si označíme. Z posloupnosti vyškrtáme všechny jeho násobky.
- Takto postupujeme, dokud je v posloupnosti nějaké neoznačené číslo.

Stačí postupovat pouze do doby, dokud není jako prvočíslo označeno číslo vyšší než $\sqrt{n}$. V takové chvíli už všechna zbývající čísla jsou nutně prvočísla.

Example

Najděte všechna prvočísla menší než 30.

Eulerova funkce

Definition

Eulerova funkce $\varphi(n) =$ počet všech čísel mezi 1 a $n-1$, která jsou nesoudělná s n.

Eulerova funkce

Definition

Eulerova funkce $\varphi(n) =$ počet všech čísel mezi 1 a $n-1$, která jsou nesoudělná s n.

Zejména platí: $\varphi(n) =$ počet invertibilních prvků v $(\mathbb{Z}_n, \cdot)$.

Eulerova funkce

Definition

Eulerova funkce $\varphi(n) =$ počet všech čísel mezi 1 a $n-1$, která jsou nesoudělná s n.

Zejména platí: $\varphi(n) =$ počet invertibilních prvků v $(\mathbb{Z}_n, \cdot)$.

Example

Najděte $\varphi(10)$.

Eulerova funkce

Definition

Eulerova funkce $\varphi(n) =$ počet všech čísel mezi 1 a $n-1$, která jsou nesoudělná s n.

Zejména platí: $\varphi(n) =$ počet invertibilních prvků v $(\mathbb{Z}_n, \cdot)$.

Example

Najděte $\varphi(10)$.

- Čísla nesoudělná s 10 jsou

Eulerova funkce

Definition

Eulerova funkce $\varphi(n) =$ počet všech čísel mezi 1 a $n-1$, která jsou nesoudělná s n.

Zejména platí: $\varphi(n) =$ počet invertibilních prvků v $(\mathbb{Z}_n, \cdot)$.

Example

Najděte $\varphi(10)$.

- Čísla nesoudělná s 10 jsou $1, 3, 7, 9$.

Eulerova funkce

Definition

Eulerova funkce $\varphi(n) =$ počet všech čísel mezi 1 a $n-1$, která jsou nesoudělná s n.

Zejména platí: $\varphi(n) =$ počet invertibilních prvků v $(\mathbb{Z}_n, \cdot)$.

Example

Najděte $\varphi(10)$.

- Čísla nesoudělná s 10 jsou $1, 3, 7, 9$.
- Tedy platí $\varphi(10) = 4$.

Vlastnosti Eulerovy funkce

Theorem

Platí následující pravidla pro výpočet $\varphi(n)$:

Vlastnosti Eulerovy funkce

Theorem

Platí následující pravidla pro výpočet $\varphi(n)$*:*

1. $\varphi(n) = n - 1$, *je-li* n *prvočíslo,*

Vlastnosti Eulerovy funkce

Theorem

Platí následující pravidla pro výpočet $\varphi(n)$*:*

1. $\varphi(n) = n - 1$, *je-li* n *prvočíslo,*
2. $\varphi(n^k) = n^{k-1} \cdot (n - 1)$, *je-li* n *prvočíslo,*

Vlastnosti Eulerovy funkce

Theorem

Platí následující pravidla pro výpočet $\varphi(n)$:

1. $\varphi(n) = n - 1$, *je-li n prvočíslo,*
2. $\varphi(n^k) = n^{k-1} \cdot (n - 1)$, *je-li n prvočíslo,*
3. $\varphi(m \cdot n) = \varphi(m) \cdot \varphi(n)$, *jsou-li m a n nesoudělná.*

Proof: ...

Vlastnosti Eulerovy funkce

Theorem

Platí následující pravidla pro výpočet $\varphi(n)$:

1. $\varphi(n) = n - 1$, *je-li n prvočíslo,*
2. $\varphi(n^k) = n^{k-1} \cdot (n - 1)$, *je-li n prvočíslo,*
3. $\varphi(m \cdot n) = \varphi(m) \cdot \varphi(n)$, *jsou-li m a n nesoudělná.*

Proof: ...

Example

$$\varphi(10) = \varphi(2 \cdot 5) = \varphi(2) \cdot \varphi(5) = 1 \cdot 4 = 4$$

Vlastnosti Eulerovy funkce

Theorem

Platí následující pravidla pro výpočet $\varphi(n)$:

1. $\varphi(n) = n - 1$, *je-li n prvočíslo,*
2. $\varphi(n^k) = n^{k-1} \cdot (n - 1)$, *je-li n prvočíslo,*
3. $\varphi(m \cdot n) = \varphi(m) \cdot \varphi(n)$, *jsou-li m a n nesoudělná.*

Proof: ...

Example

$$\varphi(10) = \varphi(2 \cdot 5) = \varphi(2) \cdot \varphi(5) = 1 \cdot 4 = 4$$

Výpočet $\varphi(n)$ závisí na znalosti rozkladu n na součin prvočísel.

Eulerova věta

Theorem

Pro libovolné $x \in \mathbb{Z}$ *a* $n \in \mathbb{N}$ *tak, že* $NSD(x, n) = 1$, *platí*

$$x^{\varphi(n)} \equiv 1 \mod n.$$

Proof: ...

Eulerova věta

Theorem

Pro libovolné $x \in \mathbb{Z}$ *a* $n \in \mathbb{N}$ *tak, že* $NSD(x, n) = 1$*, platí*

$$x^{\varphi(n)} \equiv 1 \mod n.$$

Proof: ...

- Dostáváme alternativní cestu ke hledání inverzí v $(\mathbb{Z}_n, \cdot)$.
- Inverzní prvek k prvku $[a]_n \in \mathbb{Z}_n$ je prvek $[a^{\varphi(n)-1}]_n$.

Příklad

- Najděte inverzi prvku $[35]_{37}$ v $(\mathbb{Z}_{37}, \cdot)$.

Příklad

- Najděte inverzi prvku $[35]_{37}$ v $(\mathbb{Z}_{37}, \cdot)$.
- Najděte inverzi prvku $[7]_{10}$ v $(\mathbb{Z}_{10}, \cdot)$.

Fermatova věta

Jako důsledek Eulerovy věty dostaneme následující tvrzení známé jako (malá) Fermatova věta:

Theorem

Pro libovolné $x \in \mathbb{Z}$ a n prvočíslo tak, že $NSD(x, n) = 1$, platí

$$x^{n-1} \equiv 1 \mod n.$$

Proof: ...

1 Prvočísla

2 Kongruenční rovnice

3 Testování prvočíselnosti a faktorizace

4 Šifrovací algoritmus RSA

Lineární kongruence o jedné neznámé

Theorem

- *Mějme lineární kongruenci*

$$ax \equiv b \mod n.$$

Tato kongruence má řešení právě tehdy, když

$$NSD(a, n) \mid b.$$

- *Pokud toto platí, pak rovnice má právě $NSD(a, n)$ řešení modulo n.*

Proof: ...

Lineární kongruence o jedné neznámé

- Pokud $NSD(a, n) \nmid b$, rovnice nemůže mít řešení.
- Pokud $NSD(a, n) \mid b$ a kongruence má víc než jedno řešení, t.j. $NSD(a, n) > 1$, vydělíme nejprve kongruenci číslem $NSD(a, n)$. Pak bude mít takto upravená kongruence jedno řešení.
- Využitím Euklidova algoritmu, Bezoutovy rovnosti a vhodnými úpravami vyřešíme upravenou kongruenci. Pomocí jejího řešení získáme řešení původní kongruence.

Nějaké příklady

- Řešte lineární kongruenci o jedné neznámé

$$4x \equiv 3 \mod 7.$$

Nějaké příklady

- Řešte lineární kongruenci o jedné neznámé

$$4x \equiv 3 \mod 7.$$

- Řešte lineární kongruenci o jedné neznámé

$$21x \equiv 6 \mod 9.$$

Soustavy lineárních kongruencí

Theorem

- *Systém dvou kongruencí*

$$x \equiv c_1 \mod n_1,$$
$$x \equiv c_2 \mod n_2$$

má řešení právě tehdy, když

$$c_1 \equiv c_2 \mod NSD(n_1, n_2).$$

- *Pokud toto platí, pak existuje právě jedno c takové, že soustava je ekvivalentní s kongruencí*

$$x \equiv c \mod nsn(n_1, n_2).$$

Soustavy lineárních kongruencí

- První kongruenci rozepíšeme z definice zbytkové třídy a dosadíme do druhé kongruence.
- Druhou kongruenci vyřešíme a přípustné hodnoty dosadíme do první kongruence.

Postup při řešení více kongruencí je obdobný.
Protože každou lineární kongruenci lze převést do tvaru se separovanou neznámou jejím vyřešením (nebo konstatováním, že řešení nemá), dostáváme metodu k řešení libovolného systému lineárních kongruencí s jednou neznámou.

Nějaký příklad

Řešte systém kongruencí:

$$x \equiv -5 \mod 20$$
$$x \equiv 2 \mod 13$$

1 Prvočísla

2 Kongruenční rovnice

3 Testování prvočíselnosti a faktorizace

4 Šifrovací algoritmus RSA

Co se zatím ví?

Testování prvočíselnosti ... rozhodnutí, zda je dané číslo n prvočíslo:

- Můžeme zkoušet dělitelnost čísla n všemi poteciálními děliteli. Prakticky je to ovšem nerealizovatelné, protože takový algoritmus je exponenciální.
- Existují a v praxi se používají polynomiální pravděpodobnostní algoritmy. Příkladem jsou
  - Fermatův test,
  - Rabinův-Millerův test.

  (Pravděpodobnostní=rozhodují správně s určitou pravděpodobností.)
- Teprve od roku 2002 je znám polynomiální algoritmus. Pracuje v čase $\mathcal{O}(n^{6.5})$. Autory jsou prof. Manindra Agrawal a jeho studenti Neeraj Kayal a Nitin Saxena.

Fermatův test

Rozhodujeme, zda číslo n je prvočíslo:

1. Náhodně zvolíme celé číslo x mezi 2 a $n-1$.
2. Spočteme $x^{n-1} \mod n$.
3. Pokud vyjde 1, prohlásíme n za prvočíslo, jinak za číslo složené.

Fermatův test

Rozhodujeme, zda číslo n je prvočíslo:

1. Náhodně zvolíme celé číslo x mezi 2 a $n-1$.
2. Spočteme $x^{n-1} \mod n$.
3. Pokud vyjde 1, prohlásíme n za prvočíslo, jinak za číslo složené.

Připomeňme pro úplnost malou Fermatou větu:

Theorem

Pro libovolné $x \in \mathbb{Z}$ a n prvočíslo tak, že $NSD(x, n) = 1$, platí

$$x^{n-1} \equiv 1 \mod n.$$

Analýza Fermatova testu

1. Fermatův test vždy prohlásí prvočíslo za prvočíslo. Složené číslo prohlásí buď za složené číslo nebo za prvočíslo.

Analýza Fermatova testu

1. Fermatův test vždy prohlásí prvočíslo za prvočíslo. Složené číslo prohlásí buď za složené číslo nebo za prvočíslo.
2. Složené číslo prohlásí za složené, pokud najde x takové, že neplatí $x^{n-1} \equiv 1 \mod n$. Takové číslo x se nazývá *Fermatův svědek* složenosti čísla n.

Analýza Fermatova testu

1. Fermatův test vždy prohlásí prvočíslo za prvočíslo. Složené číslo prohlásí buď za složené číslo nebo za prvočíslo.
2. Složené číslo prohlásí za složené, pokud najde x takové, že neplatí $x^{n-1} \equiv 1 \mod n$. Takové číslo x se nazývá *Fermatův svědek* složenosti čísla n.
3. Průšvih: Existují složená čísla, která žádného Fermatova svědka nemají, tzv. *Carmichaelova čísla*.

Analýza Fermatova testu

1. Fermatův test vždy prohlásí prvočíslo za prvočíslo. Složené číslo prohlásí buď za složené číslo nebo za prvočíslo.
2. Složené číslo prohlásí za složené, pokud najde x takové, že neplatí $x^{n-1} \equiv 1 \mod n$. Takové číslo x se nazývá *Fermatův svědek* složenosti čísla n.
3. Průšvih: Existují složená čísla, která žádného Fermatova svědka nemají, tzv. *Carmichaelova čísla*.
4. Lze ukázat: Pokud n není ani prvočíslo ani Carmichaelovo číslo, platí $x^{n-1} \equiv 1 \mod n$ pro nejvýše $\frac{n}{2}$ různých x.

Rabinův - Millerův test

1. Náhodně zvolíme celé číslo x mezi 2 a $n-1$.
2. Pokud $NSD(x,n) \neq 1$, označíme n za složené.
3. Najdeme t a liché m taková, že $n-1 = 2^t \cdot m$.
4. Spočítáme $b_0 \equiv x^m \mod n$.
5. Spočítáme $b_1, \ldots, b_t$ tak, že $b_{i+1} \equiv b_i^2 \mod n$. (Pak $b_t \equiv x^{n-1} \mod n$.)
6. Pokud neplatí $b_t \equiv 1 \mod n$, pak n je složené (a x je toho Fermatův svědek).
7. Pokud platí $b_i \equiv 1 \mod n$ pro $i = 0, \ldots t$, pak prohlásíme n za prvočíslo.
8. Jinak vezmeme nejvyšší i, pro něž neplatí $b_i \equiv 1 \mod n$. Pokud je $b_i \equiv -1 \mod n$, označíme n za prvočíslo, jinak za číslo složené (a x je toho *Riemannův svědek*).

Analýza Rabinova - Millerova testu

Theorem

Rabinův - Millerův test testuje prvočíselnost v polynomiálním čase. Na prvočísla odpovídá správně a složená čísla prohlašuje za prvočísla s pravděpodobností $\frac{1}{4}$.

Co se zatím ví?

Faktorizace čísla n ... rozklad čísla n na součin prvočísel:

- Není známo, zda polynomiální algoritmus existuje.
- Toho využívá mnoho šifrovacích metod.

1 Prvočísla

2 Kongruenční rovnice

3 Testování prvočíselnosti a faktorizace

4 Šifrovací algoritmus RSA

RSA algoritmus

- Algorirtmus používá teorie zbytkových tříd a kongruencí pro šifrování a dešifrování zpráv.

RSA algoritmus

- Algorirtmus používá teorie zbytkových tříd a kongruencí pro šifrování a dešifrování zpráv.
- Zkratka pochází z iniciál autorů, kterými jsou Ronald Linn Rivest, Adi Shamir a Leonard Max Adleman.

RSA algoritmus

- Algorirtmus používá teorie zbytkových tříd a kongruencí pro šifrování a dešifrování zpráv.
- Zkratka pochází z iniciál autorů, kterými jsou Ronald Linn Rivest, Adi Shamir a Leonard Max Adleman.
- Jedná se o *asymetrický šifrovací systém* ... k zašifrování a dešifrování se používají dva odlišné klíče.

Algoritmus funguje následujícím způsobem:

(1) Vezmeme dvě dostatečně velká prvočísla p a q.

(2) Položíme $n = p \cdot q$.

(3) Najdeme Eulerovu funkci $\varphi(n)$, což je se znalostí p a q snadné. (Bez jejich znalosti je to neřešitelné!) Dostaneme

$$\varphi(n) = (p-1) \cdot (q-1).$$

(4) Zvolíme číslo r, které je menší než $\varphi(n)$ a je s ním nesoudělné.
Ekvivalentně: Zvolíme r menší než $\varphi(n)$, které reprezentuje invertibilní zbytkovou třídu modulo $\varphi(n)$.

(5) Najdeme číslo s, které řeší kongruenci

$$r \cdot s \equiv 1 \mod \varphi(n).$$

Ekvivalentně: Najdeme číslo s, které reprezentuje zbytkovou třídu inverzní ke zbytkové třídě r modulo $\varphi(n)$.

Tímto získáme následující data:

- *veřejný klíč... dvojice čísel* (n, r)
- *privátní klíč... dvojice čísel* (n, s)

Algoritmus funguje následujícím způsobem:

Zprávu, kterou chceme zašifrovat, reprezentujeme jako číslo z menší než n, které je nesoudělné s n.

- Zprávu z zašifrujeme tak, že ji umocníme na veřejný klíč r modulo n. Výsledek je šifra w. Tedy máme

$$w \equiv z^r \mod n.$$

- Šifru w můžeme dešifrovat tak, že ji umocníme na privátní klíč s modulo n. Tedy tvrdíme:

$$z \equiv w^s \mod n.$$

Proč to funguje?

- Podle předpokladů máme

$$r \cdot s = 1 + k \cdot \varphi(n)$$

pro vhodné $k \in \mathbb{Z}$. Pak modulo n dostaneme

$$w^s \equiv (z^r)^s \equiv z^{1+k\cdot\varphi(n)} \equiv (z^{\varphi(n)})^k \cdot z \equiv z.$$

- I když známe veřejný klíč (n, r) a šifru w, nejsme schopni nalézt s a tedy ani z. Algoritmus je založen zejména na tom, že je mnohem lehčí nalézt prvočísla p, q než rozložit jejich součin n.