Algoritmizace, automaty, prvočísla, modulární aritmetika.

Matematické algoritmy (K611MA)

Jan Přikryl

Ústav aplikované matematiky
ČVUT v Praze, Fakulta dopravní

1. konzultace K611MA
pátek 7. října 2010

verze: 2010-10-08 09:45

Obsah konzultace

Organizační informace
Aneb kde hledat další informace

Přednášející: Dr. Jan Přikryl
E-mail: prikryl@fd.cvut.cz
Místnost: F407
Konzultace: po předchozí dohodě mailem

Informace o náplni a průběhu přednášek předmětu, zkoušce, PDF přednášek, odkazy na web a na doplňkovou literaturu, zadání úkolů a další naleznete na adrese

http://zolotarev.fd.cvut.cz/ma/

Obsah konzultace

Vývojová stádia algoritmu

Aneb intuitivní stádia geneze

V literatuře lze narazit na tvrzení, že jakékoliv řešení problému spadá do některé z následujících kategorií:

1. Na první pohled zřejmý postup
2. Metodický postup
3. Chytrý postup
4. Geniální nápad

Celkově jde o postupný vývoj „inteligence“ algoritmu od většinou naivního a časově neefektivního postupu ke složitějším, ale efektivnějším metodám řešení.

Vývojová stádia algoritmu

Aneb intuitivní stádia geneze

V literatuře lze narazit na tvrzení, že jakékoliv řešení problému spadá do některé z následujících kategorií:

1. Na první pohled zřejmý postup
2. Metodický postup
3. Chytrý postup
4. Geniální nápad

Celkově jde o postupný vývoj „inteligence“ algoritmu od většinou naivního a časově neefektivního postupu ke složitějším, ale efektivnějším metodám řešení.

Vývojová stádia algoritmu

Aneb intuitivní stádia geneze

V literatuře lze narazit na tvrzení, že jakékoliv řešení problému spadá do některé z následujících kategorií:

1. Na první pohled zřejmý postup
2. Metodický postup
3. Chytrý postup
4. Geniální nápad

Celkově jde o postupný vývoj „inteligence“ algoritmu od většinou naivního a časově neefektivního postupu ke složitějším, ale efektivnějším metodám řešení.

Vývojová stádia algoritmu

Aneb intuitivní stádia geneze

V literatuře lze narazit na tvrzení, že jakékoliv řešení problému spadá do některé z následujících kategorií:

1. Na první pohled zřejmý postup
2. Metodický postup
3. Chytrý postup
4. Geniální nápad

Celkově jde o postupný vývoj „inteligence“ algoritmu od většinou naivního a časově neefektivního postupu ke složitějším, ale efektivnějším metodám řešení.

Vývojová stádia algoritmu
Primitivní a naivní řešení

- Základní úroveň pokusů problém uchopit, pokud se nevyznáme (nebo jsme líní)
- Prohledávání příliš rozsáhlého prostoru možných řešení
- Většinou je to to, co vás napadne jako první ;-).

Příklad

Řazení seznamu záměnou sousedních prvků (bubble-sort) či zatřiďováním vždy jednoho prvku na správnou pozici (insert-sort, ten má ovšem své opodstatnění).

Vývojová stádia algoritmu
Metodická řešení

- Hlubší analýza problému
- Netriviální metodický postup
- Rozdělení úlohy na podproblémy

Příklad

Řazení seznamu rekurzivní metodou merge-sort, respektive quick-sort. Možná heap-sort, i když ten aspoň zčásti patří i na další stránku.

Vývojová stádia algoritmu
Chytrá řešení

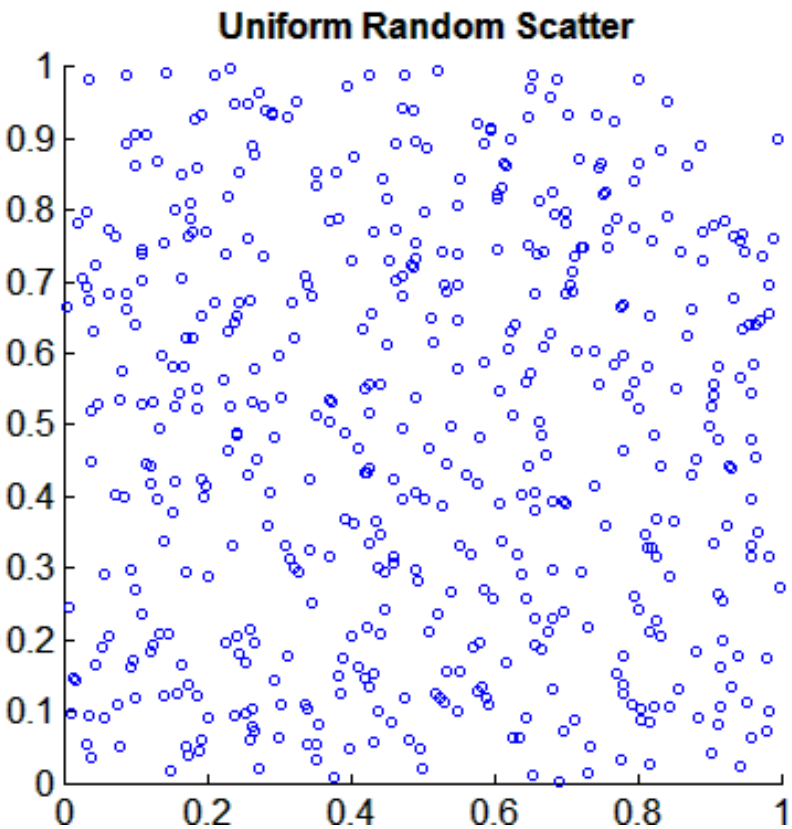
- Často specializovaná
- Vyžadují už jít do hloubky zpracovávaného problému, analyzovat vlastnosti dat
- Většinou není na první pohled jasné, proč to funguje
- Lehká nebývá ani analýza složitosti

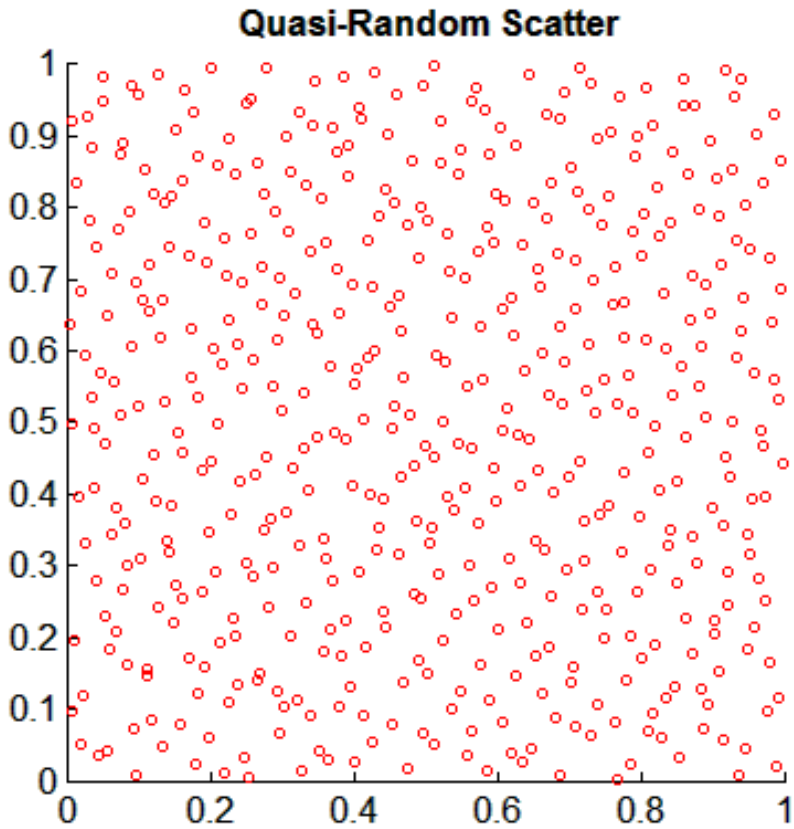
Příklad

Snad záměna pseudonáhodných za kvazináhodná čísla pro generování vzorků v Monte-Carlo simulacích. Nebo sekvenční Monte-Carlo.

Odbočka

Rozdíl mezi pseudonáhodným a kvazináhodným





Rozdělení algoritmů podle způsobu implementace

Když dva dělají totéž, nemusí to být stejným způsobem

Existuje mnoho různých možných přístupů k algoritmizaci postupu řešení:

1. Rekurzivní vs. iterativní přístup
2. Logické programování (algoritmus=logika+řízení)
3. Sériový vs. paralelní výpočet
4. Deterministický vs. nedeterministický přístup
5. Přesné vs. přibližné řešení

Rozdělení algoritmů podle způsobu implementace

Když dva dělají totéž, nemusí to být stejným způsobem

Existuje mnoho různých možných přístupů k algoritmizaci postupu řešení:

1. Rekurzivní vs. iterativní přístup
2. Logické programování (algoritmus=logika+řízení)
3. Sériový vs. paralelní výpočet
4. Deterministický vs. nedeterministický přístup
5. Přesné vs. přibližné řešení

Rozdělení algoritmů podle způsobu implementace

Když dva dělají totéž, nemusí to být stejným způsobem

Existuje mnoho různých možných přístupů k algoritmizaci postupu řešení:

1. Rekurzivní vs. iterativní přístup
2. Logické programování (algoritmus=logika+řízení)
3. Sériový vs. paralelní výpočet
4. Deterministický vs. nedeterministický přístup
5. Přesné vs. přibližné řešení

Rozdělení algoritmů podle způsobu implementace

Když dva dělají totéž, nemusí to být stejným způsobem

Existuje mnoho různých možných přístupů k algoritmizaci postupu řešení:

1. Rekurzivní vs. iterativní přístup
2. Logické programování (algoritmus=logika+řízení)
3. Sériový vs. paralelní výpočet
4. Deterministický vs. nedeterministický přístup
5. Přesné vs. přibližné řešení

Rozdělení algoritmů podle způsobu implementace

Když dva dělají totéž, nemusí to být stejným způsobem

Existuje mnoho různých možných přístupů k algoritmizaci postupu řešení:

1. Rekurzivní vs. iterativní přístup
2. Logické programování (algoritmus=logika+řízení)
3. Sériový vs. paralelní výpočet
4. Deterministický vs. nedeterministický přístup
5. Přesné vs. přibližné řešení

Rozdělení algoritmů podle způsobu implementace

Rekurzivní vs. iterativní přístup

Rekurze: při řešení úlohy provádí algoritmus sám sebe nad vybranou podmnožinou vstupních dat.

Iterace: při řešení úlohy provádí algoritmus opakovaně (v cyklu) stejnou úlohu nad měnící se množinou dat.

Example

Výpočet Fibonacciho posloupnosti $F(n) = F(n-1) + F(n-2)$ lze provádět oběma způsoby.

Rozdělení algoritmů podle způsobu implementace

Logické programování (algoritmus=logika+řízení)

Použití prostředků matematické logiky k vyjádření postupu výpočtu.

Logický program je dán

- výrokovou logikou – zápis odvozovacích pravidel
- řízením – zpracování zapsaných pravidel nějakou formou automatického dokazování

Příklad **deklarativního programování:** výroková logika pouze říká, *jaký* výpočet se má provést a nikoliv, *jak* tento výpočet provést.

Rozdělení algoritmů podle způsobu implementace

Sériový vs. paralelní výpočet

Při sériovém výpočtu počítáme vždy jenom jednu instrukci, algoritmus se vykonává jedinkrát. *Varianta:* SISD.

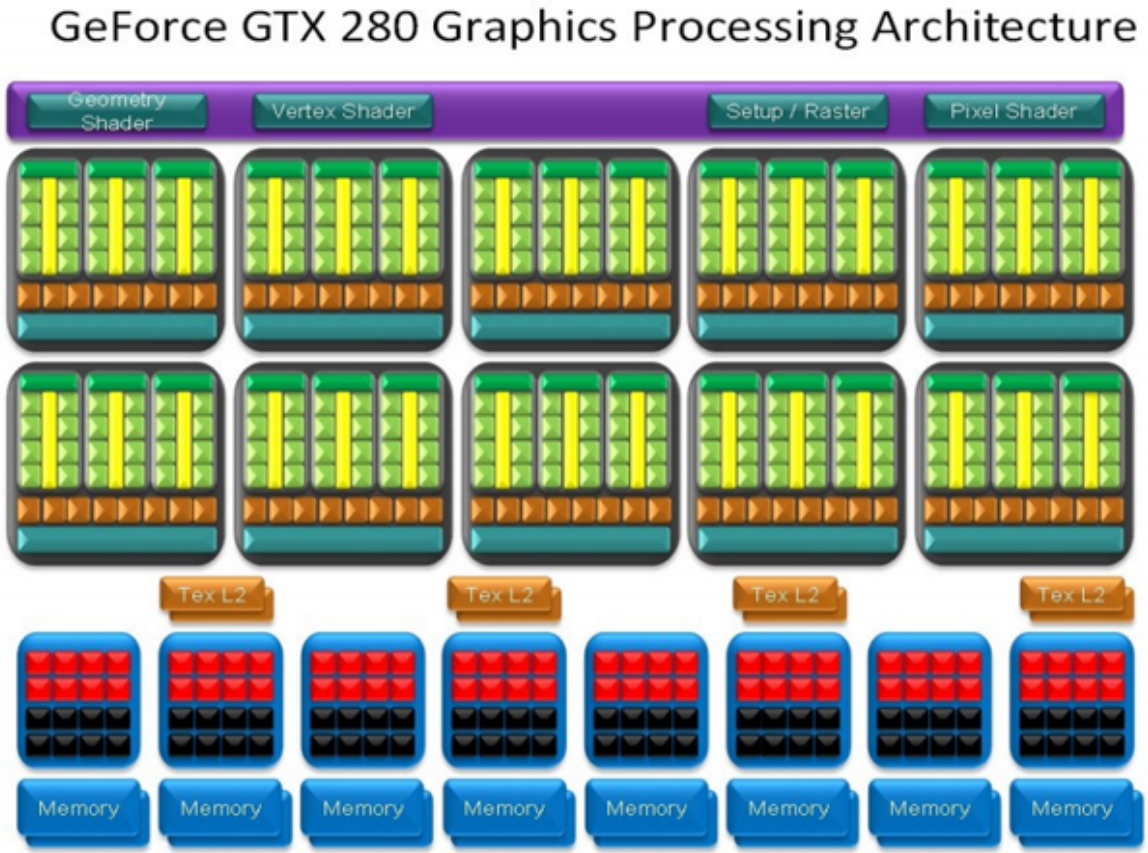
Paralelní výpočet znamená více instancí algoritmu na více procesorech nebo počítačích. *Varianty:* MISD, SIMD, MIMD.

Example

SIMD: GPU moderních grafických karet.
MIMD: distribuované výpočty Seti@home, GIMPS a podobné.

Výhoda paralelního přístupu

Jak vypadá vevnitř grafická karta



Celkem 10 ALU skupin, v každé 3×8 ALU jednotek (3 stínovací procesory a 8 texturovacích jednotek v jedné skupině).

Rozdělení algoritmů podle způsobu implementace

Deterministický vs. nedeterministický přístup

Deterministický přístup odpovídá striktní definici algoritmu: vždy dospěje ke stejnému výsledku.

Nedeterministický algoritmus obsahuje jeden nebo více bodů, v nichž následující krok není pevně určen a náhodně se zvolí jedna z předem definovaných akcí – nedospěje vždy ke stejnému výsledku.

Example

Balení (mého) batohu na hory.

Rozdělení algoritmů podle způsobu implementace

Přesné vs. přibližné řešení

Některé úlohy trvají vyřešit příliš dlouho nebo je nelze vyřešit v rozumném čase vůbec.

Pro „těžké“ problémy je často postačující alespoň přibližné (téměř optimální) řešení.

Example

Problém obchodního cestujícího: Zkuste si zoptimalizovat cestu přes stovku zastávek pro kurýra UPS/DHL/FedExu.

Základní postupy při algoritmizaci

Aneb rozdělení podle návrhového paradigmatu

1. Rozděl a panuj
2. Redukce
3. Lineární programování
4. Dynamické programování
5. Hladové algoritmy (greedy methods)
6. Probabilistické a heuristické algoritmy

Rozděl a panuj
Divide and Conquer

Klasický přístup k dekompozici problému na jednodušší podúlohy stejného problému (divide) a spojení jejich řešení v jeden celek (conquer).

- Často spojen s rekurzí
- Podproblémy nemusí být nutně *zcela* stejného typu
- *Výhody:* snadná paralelizace, rychlost, vhodný pro konceptuálně složité problémy
- *Nevýhody:* paměťová náročnost, rekurze

Example

Již zmíněný merge-sort, jenž dělí řazená data na menší podmnožiny, jež pak řadí stejným přístupem do té doby, než narazí na jednoprvkovou množinu dat – ta je seřazená vždy.

Další příklady: FFT, metoda půlení intervalu v numerice.

Redukce

Převod do duální reprezentace

Redukcí nazýváme transformaci řešeného problémy na jiný, ekvivalentní problém, jenž umíme řešit efektivněji.

- Převádíme složité úlohy na úlohy s nižší asymptotickou složitostí
- I za cenu převodu bude řešení trvat kratší dobu

Example

Hledání mediánu množiny dat. Prvním krokem může být seřazení celé množiny do vektoru dat, druhým potom výběr prostředního prvku z tohoto vektoru.

Lineární programování
Jednoduchý přístup k optimalizaci

Optimalizační postup na nalezení váženého maxima soustavy lineárních rovnic a nerovnic za určitých pevných omezení

- Problém vyjádříme soustavou rovnic, nerovnic a omezení
- Tuto soustavu nám vyřeší nějaký generický řešič (simplexová metoda)
- Není to programovací postup per se
- Častou variantou je celočíselné programování (prostor omezen na celá čísla)

Příklady

Nalezení maximálního toku mezi dvěma uzly orientovaného grafu. Optimalizace délky zelených pro SSZ snižující délky front vozidel na semaforech.

Dynamické programování
Jak si ušetřit práci

Při řešení složitých problémů s určitou strukturou se často velké bloky podúloh počítají vícekrát. Dynamické programování zamezuje opakovanému výpočtu zapamatováním si výsledků vyřešených podproblémů

- Podobné paradigmatu rozděl a panuj – podúlohy ale jsou různého typu
- Pokud se v řešení úlohy nevyskytují opakující se shodné podúlohy, nijak nám DP nepomůže

Example

Nalezení nejkratší cesty mezi dvěma uzly orientovaného ohodnoceného grafu.

Hladové algoritmy
Greedy Algorithms

Určitou třídu úloh lze řešit výběrem lokálního optima (maxima, minima) v každém kroku algoritmu a mít zaručeno, že dojdeme ke globálnímu optimu.

Example

Výběr nejmenšího počtu platidel pro určitou sumu.
Nefunguje pro denominace 1,7,10.

Probabilistické a heuristické metody

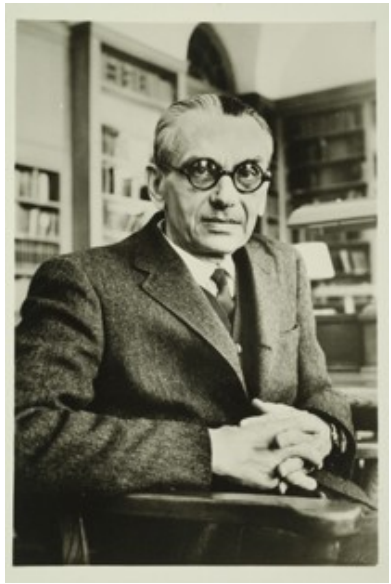
Probabilistický přístup používají **randomizované algoritmy**, například Monte-Carlo metody. Výsledkem je (rychlý) intervalový odhad řešení na určité hladině pravděpodobnosti.

Heuristické metody používají k řešení hrubý odhad na základě zkušeností programátora či odpozorovaných obvyklých výsledků.

Obsah konzultace

Matematické modely počítačů

Již v roce 1936 se *Alan Turing*, *Alonso Church* a *Kurt Gödel* zabývali teoretickými základy, na nichž by se dala moderní počítačová věda definovat.



Matematické modely počítačů

Již v roce 1936 se *Alan Turing*, *Alonso Church* a *Kurt Gödel* zabývali teoretickými základy, na nichž by se dala moderní počítačová věda definovat.

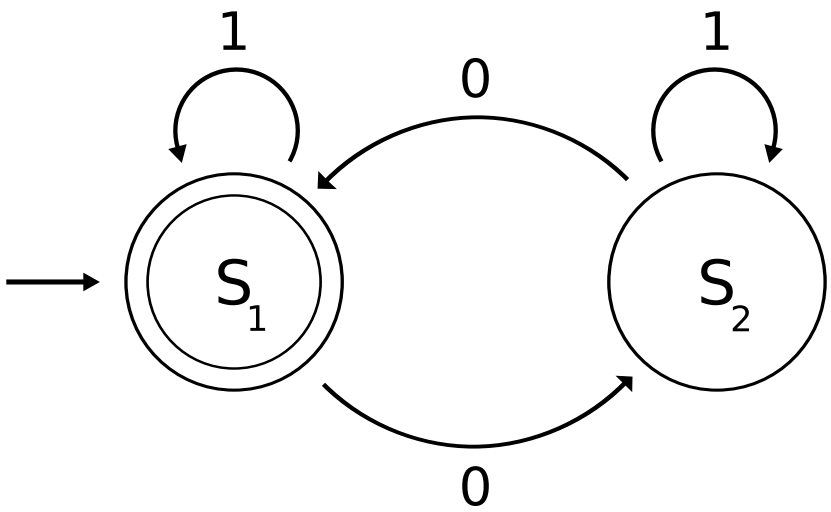
Nezávisle na sobě hledali univerzální model, který by z hlediska funkce bez ohledu na fyzikální vlastnosti popsal chování libovolného výpočtu – algoritmu.

Nejčastěji zmiňovaným řešením se stal model Alana Turinga, který byl založen na matematické abstrakci výpočetního stroje jako *konečného automatu*.

Co je to konečný automat
Finite State Machine (FSM)

Primitivní automat, jenž čte symboly nějaké definované abecedy na vstupu a podle nich přechází mezi různými vnitřními stavy. Nemá vnitřní paměť.

Může být deterministický nebo nedeterministický.



Turingův stroj
Formální popis

Motivace: Existuje mechanický proces, kterým je možno rozhodnout o pravdivosti libovolného matematického teorému nebo výroku.

Turingův stroj simuluje práci matematika při vytváření důkazu:

- nekonečná tabule (pro psaní i čtení)
- mozek (řídící jednotka)

Formalizace Turingova stroje

- místo tabule oboustranně nekonečná páska
- místo křídy čtecí a zapisovací hlava, kterou lze posouvat
- místo mozku konečná řídicí jednotka

Turingův stroj
Vlastnosti

Takový stroj měl několik základních vlastností:

- musel nahradit složitou symboliku matematických kroků. V takovém případě šlo každou konečnou množinu symbolů nahradit pouze dvěma symboly (jako je 0 a 1) a prázdnou mezerou, která by oba symboly oddělovala.
- podobně jako si matematik zapisuje poznámky na papír, má Turingův stroj k zápisu nekonečnou pásku skládající se z buněk, do/ze kterých se symbol zapisuje/čte.
- nad touto páskou je možné provádět za pomocí čtecí hlavy operace čtení, zápisu a posunu pásky (*read*, *write*, *shift left*, *shift right*).
- protože je možné symboly číst, zapisovat nebo se posunovat po pásce, je pro Turingův stroj důležitý vnitřní stav, ve kterém operaci čtení provádíme (čtený symbol a stav určují další akci a přechod do dalšího stavu)

Turingův stroj
Formální definice

Klasický Turingův stroj je konečný automat, jenž můžeme popsat sedmicí $T \equiv \{Q, \Gamma, b, \Sigma, \delta, q_0, F\}$, kde

- Q je konečná množina vnitřních stavů,
- Γ je konečná množina symbolů abecedy na pásce,
- $b \in \Gamma$ je symbol pro prázdné políčko,
- $\Sigma \subseteq \Gamma \setminus \{b\}$ je množina vstupních symbolů na pásce,
- $\delta : Q \times \Gamma \to Q \times \Gamma \times \{L, P\}$ je přechodová funkce, popisující změnu stavu, zápis na pásku a posun hlavy,
- $q_0 \in Q$ je počáteční stav stroje,
- $F \subseteq Q$ je množina koncových (akceptovaných) stavů.

Cokoliv, co odpovídá tomuto formálnímu zápisu, je Turingův stroj.

Obsah konzultace

Prvočísla
Definice

Známe dvě skupiny přirozených čísel $n \in \mathbb{N}$.

Prvočíslo

Prvočíslem nazýváme takové přirozené číslo $n \in \mathbb{N}$, které je *beze zbytku dělitelné právě dvěma různými přirozenými čísly* a to jedničkou a samo sebou.
Číslo 1 tedy není prvočíslo.

Číslo složené

Celé číslo různá od jedné, jež není prvočíslem, nazýváme **složené číslo**.

Prvočísla
Definice

Známe dvě skupiny přirozených čísel $n \in \mathbb{N}$.

Prvočíslo

Prvočíslem nazýváme takové přirozené číslo $n \in \mathbb{N}$, které je *beze zbytku dělitelné právě dvěma různými přirozenými čísly* a to jedničkou a samo sebou.
Číslo 1 tedy není prvočíslo.

Číslo složené

Celé číslo různá od jedné, jež není prvočíslem, nazýváme **složené číslo**.

Prvočísla
Definice

Známe dvě skupiny přirozených čísel $n \in \mathbb{N}$.

Prvočíslo

Prvočíslem nazýváme takové přirozené číslo $n \in \mathbb{N}$, které je *beze zbytku dělitelné právě dvěma různými přirozenými čísly* a to jedničkou a samo sebou.
Číslo 1 tedy není prvočíslo.

Číslo složené

Celé číslo různá od jedné, jež není prvočíslem, nazýváme **složené číslo**.

Prvočísla

Vlastnosti prvočísel:

- Pro prvočíslo p platí $p \mid a \cdot b \Rightarrow (p \mid a) \vee (p \mid b)$.
- Každé složené číslo lze jednoznačně vyjádřit jako součin prvočísel.

Vzorový rozklad

Například $42 = 2 \cdot 21 = 2 \cdot 3 \cdot 7$.

- Pokud p je prvočíslo a $a \in \mathbb{Z} : 0 < a < p$, pak $p \mid (a^p - a)$.
- Ke všem celým kladným číslům $a \in \mathbb{Z} : a > 0$ lze nalézt prvočíslo $p : a < p \leq 2a$.

Příklad

Nechť $a = 42$. Nerovnici $p : 42 < p \leq 84$ splňují prvočísla 43, 47, 53, 59, 61, 67, 71, 73, 79, a 83.

Prvočísla

Vlastnosti prvočísel:

- Pro prvočíslo p platí $p \mid a \cdot b \Rightarrow (p \mid a) \vee (p \mid b)$.
- Každé složené číslo lze jednoznačně vyjádřit jako součin prvočísel.

Vzorový rozklad

Například $42 = 2 \cdot 21 = 2 \cdot 3 \cdot 7$.

- Pokud p je prvočíslo a $a \in \mathbb{Z} : 0 < a < p$, pak $p \mid (a^p - a)$.
- Ke všem celým kladným číslům $a \in \mathbb{Z} : a > 0$ lze nalézt prvočíslo $p : a < p \leq 2a$.

Příklad

Nechť $a = 42$. Nerovnici $p : 42 < p \leq 84$ splňují prvočísla 43, 47, 53, 59, 61, 67, 71, 73, 79, a 83.

Prvočísla

Vlastnosti prvočísel:

- Pro prvočíslo p platí $p \mid a \cdot b \Rightarrow (p \mid a) \vee (p \mid b)$.
- Každé složené číslo lze jednoznačně vyjádřit jako součin prvočísel.

Vzorový rozklad

Například $42 = 2 \cdot 21 = 2 \cdot 3 \cdot 7$.

- Pokud p je prvočíslo a $a \in \mathbb{Z} : 0 < a < p$, pak $p \mid (a^p - a)$.
- Ke všem celým kladným číslům $a \in \mathbb{Z} : a > 0$ lze nalézt prvočíslo $p : a < p \leq 2a$.

Příklad

Nechť $a = 42$. Nerovnici $p : 42 < p \leq 84$ splňují prvočísla 43, 47, 53, 59, 61, 67, 71, 73, 79, a 83.

Prvočísla

Vlastnosti prvočísel:

- Pro prvočíslo p platí $p \mid a \cdot b \Rightarrow (p \mid a) \vee (p \mid b)$.
- Každé složené číslo lze jednoznačně vyjádřit jako součin prvočísel.

Vzorový rozklad

Například $42 = 2 \cdot 21 = 2 \cdot 3 \cdot 7$.

- Pokud p je prvočíslo a $a \in \mathbb{Z} : 0 < a < p$, pak $p \mid (a^p - a)$.
- Ke všem celým kladným číslům $a \in \mathbb{Z} : a > 0$ lze nalézt prvočíslo $p : a < p \leq 2a$.

Příklad

Nechť $a = 42$. Nerovnici $p : 42 < p \leq 84$ splňují prvočísla 43, 47, 53, 59, 61, 67, 71, 73, 79, a 83.

Prvočísla

Pěkně od začátku ...

Seznam prvočísel

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100
101	102	103	104	105	106	107	108	109	110
111	112	113	114	115	116	117	118	119	120

Uměli byste pokračovat dál?

Kolik je prvočísel?

Označme $\pi(N)$ počet prvočísel $\leq N$.

Počítejme zkusmo hustotu prvočísel ϱ_N v intervalu $\langle 1, N \rangle$:

- v desítce čísel je $\pi(10) = 4$ prvočísla, tedy

$$\varrho_{10} = \frac{\pi(10)}{10} = \frac{4}{10} = 0{,}4$$

- ve stovce čísel je $\pi(100) = 25$ prvočísel, tedy

$$\varrho_{100} = \frac{\pi(100)}{100} = \frac{25}{100} = 0{,}25$$

- v tisícovce čísel je $\pi(1000) = 168$ prvočísel, tedy

$$\varrho_{1000} = \frac{\pi(1000)}{1000} = \frac{168}{1000} = 0{,}168$$

Kolik je prvočísel?
Hustoty prvočísel

V roce 1792 si mladý C. F. Gauss všiml, že $\pi(N)$ je přibližně rovna podílu $N/\ln N$.

N	10	10^2	10^3	10^4	10^5	10^6
ϱ_N	0,400	0,250	0,168	0,123	0,096	0,078
$1/\ln N$	0,434	0,217	0,145	0,108	0,086	0,072
$N/\ln N$	4,3429	21,715	144,76	1085,7	8685,9	72382
$\pi(N)$	4	25	168	1229	9592	78498

Kolik je prvočísel?
Prvočíselná věta

Domněnka

Nejedná se o náhodný jev, při dostatečně velkém N je hustota prvočísel v intervalu $\langle 1, N \rangle$ rovna

$$\lim_{N \to \infty} \varrho_N = \frac{1}{\ln N}$$

Gaussovi bylo tehdy patnáct let. Důkaz tohoto tvrzení přišel až o 100 let později.

Prvočíselná věta

$$\lim_{N \to \infty} \frac{\pi(N)}{\frac{N}{\ln(N)}} = 1$$

Co ještě víme o prvočíslech?
Eukleidův důkaz

Prvočísel je nekonečně mnoho:

- Předpokládejme, že existuje největší prvočíslo a označme jej p_M
- Sestrojíme součin všech prvočísel až do p_M:

$$N = 2 \cdot 3 \cdot 7 \cdots p_M = \prod_{i=1}^{M} p_i$$

- Číslo $N + 1$ nemůže být dělitelné ani jedním z prvočísel p_i, jež dělí N.
- To znamená, že $N + 1$ je buď *prvočíslo*, nebo *číslo složené*, jež má ve svém rozkladu jiné prvočíslo $p_N > p_M$.
- Spolu s Eukleidem jsme dospěli ke sporu!
- Musí tedy platit, že prvočísel je nekonečně mnoho.

Eukleidův důkaz

Eukleidův důkaz je klasický existenční důkaz: Řeší pouze otázku existence nekonečné množiny prvočísel, neřeší otázku jak nalézt všechna prvočísla.

Goldbachova hypotéza

Goldbachova hypotéza říká, že každé sudé číslo větší než 2 lze vyjádřit jako součet dvou prvočísel, například

$$\begin{aligned} 8 &= 3+5 \\ 10 &= 3+7 \\ 12 &= 5+7 \\ 14 &= 3+11 \\ 16 &= 5+11 \\ 18 &= 7+11 \end{aligned}$$

Experimentálně prověřeno do hodnot 2×10^{17}

Zajímavosti
Párová prvočísla

Párová prvočísla: jejich rozdíl je 2 (například 17 a 19), největší dosud známé prvočíselné páry jsou

$$16\,869\,987\,339\,975 \cdot 2^{171960} \pm 1$$
$$100\,314\,512\,544\,015 \cdot 2^{171960} \pm 1$$

Odhalení chyby matematického koprocesoru originálního Intel Pentium P5 (*The Intel FDIV Bug*).

Zajímavosti
The Intel FDIV Bug (1)

Thomas Nicely, Lynchburg College, Virginia (1994): Numerický výpočet součtu *harmonické řady s párovými prvočísly*.

O jaké řady jde:

- harmonická řada

$$\sum_{n=1}^{\infty} \frac{1}{n} \to \infty$$

- prvočíselná harmonická řada

$$\sum_{\forall p}^{\infty} \frac{1}{p} \to \infty$$

Obě tyto řady divergují.

Zajímavosti
The Intel FDIV Bug (2)

Oproti tomu

$$\begin{aligned}\sum_{\forall p_2}^{\infty} \frac{1}{p_2} &= \frac{1}{3}+\frac{1}{5}+\frac{1}{7}+\cdots+\frac{1}{29}+\frac{1}{31}+\cdots \\ &= 1{,}902160583104\end{aligned}$$

konverguje.

V červnu 1994 Thomas Nicely obdržel po povýšení starého počítače na P5 hodnotu

$$1{,}9021605778$$

lišící se od původních výpočtů na i486 – a v říjnu oznámil chybu v FPU Pentia.

Zajímavosti
The Intel FDIV Bug (3)

Tim Coe, Vitesse Semiconductor, Southern California

$$c = \frac{4195835}{3145727} = \frac{5 \cdot 7 \cdot (2^3 \cdot 3^4 \cdot 5 \cdot 37 + 1)}{3 \cdot 2^{20} - 1} = \\ = 1{,}33382044\ldots$$

FPU v Pentiu P5 však dávala hodnotu

$$c = \frac{4195835}{3145727} = \frac{5 \times 7 \times 119881}{13 \times 241979} = 1{,}33373906\ldots$$

Chyba nastává při reprezentaci čísel typu $M_n = 2^n - 1$, což jsou tak zvaná *Mersennova čísla*.

Mersennova čísla

Definice

Marin Mersenne (1588-1648) uveřejnil ve své knize Cogitata Physica-Mathematica (1644) tvrzení, že čísla tvaru

$$2^n - 1$$

jsou prvočísly pro $n = 2, 3, 5, 7, 13, 17, 19, 31, 67, 127, 257$ a jsou čísla složená pro ostatní přirozená čísla $n \leq 257$.

Definice

Jestliže $2^n - 1$ je prvočíslo, pak se nazývá **Mersennovo prvočíslo**.

Lze dokázat, že pokud je $2^n - 1$ prvočíslo, je i n prvočíslem.

Mersennova čísla

Ověřování

Prvočíselný charakter Mersennových čísel nebylo snadné dokázat:

- Euler (1750): $2^{31} - 1$ je prvočíslo.
- Lucas (1876): $2^{127} - 1$ je prvočíslo.
- Pervouchine (1883): Mersenne zapomněl na $2^{61} - 1$.
- Powers (?) ukázal, že existují další čísla, která Mersenne neuvedl: $2^{89} - 1$ a $2^{107} - 1$.

Mersennův interval $n \leq 257$ byl úplně prozkoumán v roce 1947 a bylo dokázáno, že správné tvrzení obsahuje 12 exponentů:

$$n = 2, 3, 5, 7, 13, 17, 19, 31, 61, 89, 107, 127.$$

Mersennova čísla

Výpočet nových

K dnešnímu dni bylo nalezeno dalších 35 Mersennových prvočísel $M_{521}, M_{607}, M_{1\,279}, \ldots, M_{42\,643\,801}, M_{43\,112\,609}$.

GIMPS (The Great Internet Mersenne Prime Search)

Paralelizované hledání jehly v kupce sena:

- Distribuovaný výpočet ve volných cyklech procesoru
- Zatím poslední Mersennovo prvočíslo bylo nalezeno 6. září 2008 ve tvaru

$$2^{37\,156\,667} - 1.$$

- `http://www.mersenne.org/`

Fermatova čísla

A jejich vztah k prvočíslům

Pro nezáporné $n \geq 0$ nazýváme n-tým **Fermatovým číslem** výraz

$$F_n = 2^{2^n} + 1.$$

Je známo, že F_n je

- prvočíslem pro $0 \leq n \leq 4$ a
- číslem složeným pro $5 \leq n \leq 23$.

Fermat se původně domníval, že F_n jsou obecně prvočísla.

Jak vlastně rozhodneme, na jaké součinitele rozložit složené číslo N?

Faktorizace prvočísel

Proč to?

Základní věta artimetiky

Každé přirozené číslo větší než 1 lze jednoznačně rozložit na součin prvočísel.

Nalezení rozkladu malých čísel na prvočísla je relativně jednoduché:

Zkouška dělením

Pro výpočet prvočíselných součinitelů čísla N stačí otestovat všechna prvočísla $p_i < \sqrt{N}$. Prvočinitele získáme například použitím Eratostenova síta.

Faktorizace prvočísel

Jak faktorizovat velká prvočísla

Náročnost faktorizace výrazně roste s délkou prvočísla.

Praktické důsledky:

- (+) kryptografie (šifrování veřejným klíčem, RSA),
- (-) výpočet multiplikativních funkcí – $\forall a, b : f(ab) = f(a) \cdot f(b)$,
- (-) kompresní algoritmy.

Metody prosévání:

- Erastothenovo síto
- (Generické—Speciální) prosévání číselného pole
- Pollardova ρ-metoda
- Rozklad na řetězové zlomky

Faktorizace prvočísel

Metody

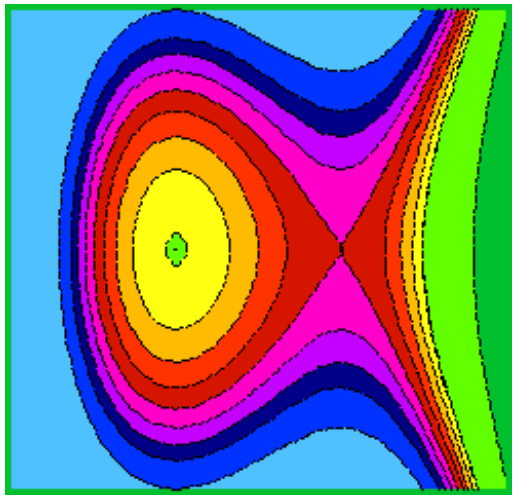
Pro rozklad velkých čísel na prvočíselné součinitele se používají celočíselné vlastnosti eliptických křivek

$$y^2 = x^3 + a\,x + b$$

Na stránkách

http://www.alpertron.com.ar/ECM.HTM

můžete ověřit účinnost těchto matematických metod.



Faktorizace Fermatových čísel

Jak na ně

- Euler (1732) nalezl rozklad

$$F_5 = 641 \cdot 6\,700\,417 = 4\,294\,967\,297.$$

- Žádné další prvočíslo tvaru $F_n = 2^{2^n} + 1$ není pro $n \geq 24$ známo.

Dějiny faktorizace

Rozklad Fermatových čísel se od doby Eulera stal velkou soutěží o vhodné algoritmy.
Samostatný mikrokosmos numerické matematiky: F_n roste v počtu cifer závratně rychle – algoritmus vhodný pro faktorizaci F_n nemusí být použitelný pro F_{n+1}.

Faktorizace Fermatových čísel

F_6

V roce 1880 Landry zveřejnil součin

$$F_6 = 274\,177 \cdot p_{14}.$$

Algoritmus, kterým Landry k tomuto výsledku dospěl, nebyl nikdy publikován.

Faktorizace Fermatových čísel

F_7

V roce 1970 Morrison a Brillhart nalezli pomocí metod řetězových zlomků součin

$$F_7 = 59\,649\,589\,127\,497\,217 \cdot p_{22}.$$

V letech 1877-1970 bylo objeveno několik nevelkých součinitelů Fermatových čísel ve tvaru $k \cdot 2^{n+2} + 1$ pro $n \geq 9$.

Western, 1903

Například již v roce 1903 Western nalezl

$$F_9 = 2\,424\,833 \times C_{148},$$

kde C_{148} je celé 148-ciferné číslo.

Faktorizace Fermatových čísel

F_8 a F_9

V roce 1980 Brent a Pollard nalezli součin

$$F_8 = 1238926361552897 \times p_{62}$$

Pollardovou ρ-metodou.

V roce 1990 skupina matematiků a počítačových odborníků kolem Pollarda použila více než 700 pracovních stanic rozmístěných po celém světě a odvodili metodou SNFS (Special Number Field Sieve) pro

$$F_9 = 2\,424\,833 \times p_{49} \times p_{99}.$$

Faktorizace Fermatových čísel

Současnost

V říjnu 2003 John Cosgrave se spolupracovníky na St. Patrick's College nalezli součinitele Fermatova číslu

$$F_{2478782} = (3 \times 2^{2478785} + 1) \cdot k.$$

Faktorizace:

- není kompletní pro všechna Fermatova čísla, o kterých víme, že jsou rozložitelná,
- například pro F_{12} není znám součinitel C_{1187} o velikosti 1187 cifer,
- podobně pro $F_{13}, F_{15}, F_{16}, F_{17}, F_{18}, F_{19}, F_{21}$ a F_{23} chybí součinitele různých ciferných délek.

Faktorizace Fermatových čísel

Přehled

n	$F_n = 2^{2^n} + 1$
0	3
1	5
2	7
3	257
4	65 537
5	$641 \cdot 6\,700\,417$
6	$274\,177 \cdot 67\,280\,421\,310\,721$
7	$59\,649\,589\,127\,497\,217 \cdot 5\,704\,689\,200\,685\,129\,054\,721$
8	$1\,238\,927\,497\,217 \cdot p_{62}$
9	$2\,424\,833 \cdot p_{49} \cdot p_{99}$

V tabulce označuje p_k k-ciferné prvočíslo. Například $F_6 = 274\,177 \cdot 67\,280\,421\,310\,721 = 274\,177 \cdot p_{14}$.

Obsah konzultace

Nejvyšší společný dělitel
Definice

V celočíselné aritmetice dělíme se zbytkem: je

$$a = qb + r.$$

Pro dvojici celých čísel a a b má smysl hledat nejvyšší celé číslo d, které obě čísla dělí beze zbytku.

Nejvyšší společný dělitel

Nejvyšší společný dělitel dvou nenulových celých čísel $a \in \mathbb{Z}$ a $b \in \mathbb{Z}$ je největší nenulové přirozené číslo $d \in \mathbb{Z} - 0$ takové, že $d|a \wedge d|b$.
Zapisujeme $\gcd(a, b) = d$.

Nesoudělná čísla

Čísla $a \in \mathbb{Z}$ a $b \in \mathbb{Z}$ nazýváme **nesoudělná** *(relative primes)*, pokud $\gcd(a, b) = 1$.

Nejvyšší společný dělitel
Definice

V celočíselné aritmetice dělíme se zbytkem: je

$$a = qb + r.$$

Pro dvojici celých čísel a a b má smysl hledat nejvyšší celé číslo d, které obě čísla dělí beze zbytku.

Nejvyšší společný dělitel

Nejvyšší společný dělitel dvou nenulových celých čísel $a \in \mathbb{Z}$ a $b \in \mathbb{Z}$ je největší nenulové přirozené číslo $d \in \mathbb{Z} - 0$ takové, že $d|a \wedge d|b$.
Zapisujeme $\gcd(a, b) = d$.

Nesoudělná čísla

Čísla $a \in \mathbb{Z}$ a $b \in \mathbb{Z}$ nazýváme **nesoudělná** *(relative primes)*, pokud $\gcd(a, b) = 1$.

Euklidova čísla

Euklidova čísla

Čísla definovaná rekurencí

$$e_n = e_1\, e_2\, e_3 \ldots e_{n-1} + 1$$

nazýváme **Euklidova čísla**.

První čtyři Euklidova čísla

$$\begin{aligned} e_1 &= 1 + 1 = 2 \\ e_2 &= 2 + 1 = 3 \\ e_3 &= 2 \times 3 + 1 = 7 \\ e_4 &= 2 \times 3 \times 7 + 1 = 43 \end{aligned}$$

jsou prvočísla.

Euklidova čísla
Pokračování

Další Euklidova čísla až na e_6

$$e_5 = 2 \cdot 3 \cdot 7 \cdot 43 + 1 = 1\,807 = 13 \cdot 139 \quad (1)$$
$$e_6 = 2 \cdot 3 \cdot 7 \cdot 43 \cdot 1\,807 + 1 = 3\,263\,443 \quad (2)$$
$$e_7 = 547 \cdot 607 \times 1\,033 \cdot 31\,051 \quad (3)$$
$$e_8 = 29\,881 \cdot 67\,003 \cdot 9\,119\,521 \cdot 6\,212\,157\,481 \quad (4)$$

jsou složená čísla. Pro všechna čísla $e_9 \ldots e_{17}$ je dokázáno, že jsou to složená čísla.

Fact

Euklidova čísla jsou **nesoudělná čísla**, *protože jejich největší společný dělitel je roven 1:*

$$\gcd(e_m, e_n) = 1.$$

Euklidův algoritmus

Chytré řešení problému

Původně formulován geometricky cca 300 př.n.l. Eukleidés hledal nejdelší úsečku, která by se beze zbytku vešla do dvou delších úseček.

Důkaz

1. Nechť a a b jsou nenulová celá čísla, jejichž gcd() počítáme.
2. Pokud $a > b$, platí $a = qb + r$.
3. Pokud existuje d takové, že $d|a$ a $d|b$, pak také $d|r$, protože pro $a = sd$ a $b = td$ bude $r = (s - qt)d$.
4. Je tedy $\gcd(a, b) = \gcd(b, r)$ a stačí tedy hledat $\gcd(b, r)$.
5. Protože $r < b$, výpočet v konečném počtu kroků skončí stavem $r = 0$.

Obsah konzultace

O čem to vlastně všechno je
Krátká motivační vložka

Modulární aritmetika je aritmetikou na množině celých čísel $\mathbb{Z}$ v níž se čísla opakují po dosažení určité hodnoty n, již nazýváme **modul**.

Na rozdíl od běžných celočíselných operací se zde po každé operaci provede ještě celočíselné dělení modulem n a výsledkem operace je zbytek po tomto dělení.

Příklad

V modulární aritmetice modulo 7 mají čísla 8 a 71 shodné reprezentace, protože $8 \bmod 7 = 1$ a zároveň $71 \bmod 7 = 1$.

K čemu to vlastně všechno je
Krátká motivační vložka

Celočíselná aritmetika v počítačích je modulární.

Příklad pro osmibitová čísla

250+10 je v osmibitové aritmetice rovno 4 (tedy $260 \bmod 2^8$).
12-16 je v osmibitové aritmetice rovno 252 (což je $-4 \bmod 2^8$).

Praktické aplikace modulární aritmetiky:

- **přenos zpráv** – ochrana zpráv proti chybám, komprese, zajištění integrity, utajování,
- **výpočetní technika** – hašovací funkce, pseudonáhodná čísla, dvojková komplementární reprezentace celých čísel, aritmetika s VELKÝMI celými čísly.

Obsah konzultace

Dělitelnost
Připomenutí

Na množině celých čísel $\mathbb{Z}$ mějme definována dvě čísla: a, b. Říkáme, že a **dělí** b, pokud existuje libovolné $c \in \mathbb{Z}$ takové, že $b = ac$.

Pro zkrácený zápis toho vztahu používáme symbol $a|b$.

Pro **společný dělitel** c čísel a a b platí, že $c|a$ a zároveň $c|b$.

Dělitelnost
Největší společný dělitel

Číslo d označujeme jako **největšího společného dělitele** čísel a a b a zapisujeme $d = \gcd(a, b)$, pokud platí, že

- číslo d je společný dělitel a a b, a
- pokud existuje $c \neq d$ takové, že $c|a$ a zároveň $c|b$, pak také $c|d$.

Číslo $\gcd(a, b)$ je tedy největším kladným celým číslem jež dělí jak a, tak i b, s výjimkou $\gcd(0, 0) = 0$.

Kongruence
Připomenutí

Uvažujme libovolný modul n takový, že $n \in \mathbb{N}$ a zvolme si dvě celá čísla $a, b \in \mathbb{Z}$.

Pokud v modulární aritmetice platí, že $a \bmod n$ a $b \bmod n$ jsou si rovny (mají stejný zbytek po dělení n), říkáme, že že a je kongruentní s b modulo n a zapisujeme

$$a \equiv b \pmod{n}.$$

Příklad

Je tedy $8 \equiv 71 \pmod{7}$, 8 je kongruentní s 71 modulo 7.

Pozor na záporná čísla: $-1 \equiv 7 \pmod{8}$.

Kongruence
Příklady

Mějme abecedu velkých písmen české abecedy, $\{\mathrm{A}, \mathrm{Á}, \mathrm{B}, \ldots, \mathrm{Z}, \mathrm{Ž}\}$, reprezentovanou numerickými hodnotami $\{1, 2, \ldots, 42\}$. Nad touto abecedou provádíme všechny matematické operace modulárně, s modulem 42.

V takové modulární aritmetice jsou si rovny například reprezentace celých čísel -41, 43 a 320328919, protože zbytek po dělení 42 je vždy 1:

$$\begin{aligned} -41 \equiv 43 \,(\mathrm{mod}\ 42) &\quad\Leftrightarrow\quad -41 = 43 + 42 \cdot (-2), \\ -41 \equiv 320328919 \,(\mathrm{mod}\ 42) &\quad\Leftrightarrow\quad -41 = 320328919 + 42 \cdot (-7626880), \\ 320328919 \equiv 43 \,(\mathrm{mod}\ 42) &\quad\Leftrightarrow\quad 320328919 = 43 + 42 \cdot 7626878. \end{aligned}$$

Znak A může tedy reprezentovat libovolné z čísel -41, 43 a 320328919.

Kongruence
Příklady

Množinu všech celých čísel, která jsou kongruentní s nějakým m modulo n je zvykem nazývat **třída kongruence** a zapisovat ji $\overline{m}$, bez uvedení modulu kongruence.

Příklad

Například číslo 3 v modulu 5 může zastupovat i všechna čísla s ním kongruentní $(\ldots, -7, -2, 3, 8, 13, \ldots)$. V textech bude tato třída kongruence označována jako $\overline{3}$.

Vlastnosti kongruence modulo n umožňují počítat pouze se zbytky po dělení tímto modulem a výsledek pak zobecnit na všechna čísla.

Obsah konzultace

Vlastnosti modulární aritmetiky

Uzavření

Modulární aritmetika je uzavřená vůči operacím sčítání a násobení:

$$\begin{aligned} \overline{a} + \overline{b} &= \overline{a+b}, \\ \overline{a} - \overline{b} &= \overline{a-b}, \\ \overline{a} \cdot \overline{b} &= \overline{a \cdot b}. \end{aligned}$$

Příklady

V aritmetice modulo 7 by mělo platit $\overline{2} + \overline{6} = \overline{1}$. Pro $9 \in \overline{2}$ a $-1 \in \overline{6}$ je výsledek $9 - 1 = 8 \in \overline{1}$.

Podobně zkuste v aritmetice modulo 7 ověřit $\overline{2} \cdot \overline{6} = \overline{5}$.

Vlastnosti modulární aritmetiky

Komutativita a asociativita

Sčítání a násobení v modulární aritmetice je komutativní a asociativní:

$$\begin{aligned} \overline{a} + \overline{b} &= \overline{b} + \overline{a}, \\ \overline{a} \cdot \overline{b} &= \overline{b} \cdot \overline{a}, \\ \left(\overline{a} + \overline{b}\right) + \overline{c} &= \overline{a} + \left(\overline{b} + \overline{c}\right), \\ \left(\overline{a} \cdot \overline{b}\right) \cdot \overline{c} &= \overline{a} \cdot \left(\overline{b} \cdot \overline{c}\right). \end{aligned}$$

Vlastnosti modulární aritmetiky

Komutativita

Pro sčítání a násobení v modulární aritmetice existuje identita, pro sčítání i inverze:

$$\begin{aligned} \overline{0} + \overline{a} &= \overline{a}, \\ \overline{a} + \overline{-a} &= \overline{0}, \\ \overline{1} \cdot \overline{a} &= \overline{a}. \end{aligned}$$

Příklady

V modulární aritmetice modulo 7 je $28 \in \overline{0}$ a $15 \in \overline{1}$. Pro jejich součet platí $(28 + 15) \bmod 7 = 43 \bmod 7 = 1$.

V modulární aritmetice modulo 3 je $10 \in \overline{1}$ a $8 \in \overline{2}$. Pro jejich součin platí $(10 \cdot 8) \bmod 3 = 80 \bmod 3 = 2$.

Jak dopadne součet 57 a -73 v aritmetice modulo 8?

Vlastnosti modulární aritmetiky

Modulární dělení

Pokud

$$a \cdot d \equiv b \cdot d \,(\mathrm{mod}\, n)$$

obecně neplatí, že také

$$a \equiv b \,(\mathrm{mod}\, n).$$

Jsou dvě varianty

1. Pro d a n nesoudělná je opravdu $a \cdot d \equiv b \cdot d \,(\mathrm{mod}\, n)$.
2. Pro $d \neq 0$ je $a \cdot d \equiv b \cdot d \,(\mathrm{mod}\, n \cdot d)$.

Vlastnosti modulární aritmetiky

Příklady na modulární dělení

Modulární dělení pro d a n nesoudělná

Pro $170 \equiv 35 \pmod 3 \longrightarrow 5 \cdot 34 \equiv 5 \cdot 7 \pmod 3$ je $34 \equiv 7 \pmod 3$, protože 3 a 5 jsou nesoudělná čísla.

Modulární dělení pro obecné $d \neq 0$

Z kongruence $10 \equiv 6 \pmod 4 \longrightarrow 5 \cdot 2 \equiv 3 \cdot 2 \pmod{2 \cdot 2}$ plyne $5 \equiv 3 \pmod 2$.

Co vyjde pro $10 \equiv 6 \pmod 3$?

Obsah konzultace

Definice

Pierre de Fermat, 1640

Pro $a \in \mathbb{Z}$ a prvočíslo $p \in \mathbb{N}$ takové, že $p \nmid a$ platí

$$a^{p-1} \equiv 1 \,(\mathrm{mod}\, p)$$

a v alternativním tvaru

$$a^{p} \equiv a \,(\mathrm{mod}\, p)$$

Ve skutečnosti je $a^{\phi(p)} \equiv 1 (\mathrm{mod}\, p)$, kde $\phi(p)$ je takzvaná **Eulerova funkce**.

Malá Fermatova věta je základním stavebním kamenem algoritmu generování šifrovacího klíče asymetrické šifry RSA. Je také nutnou podmínkou pro prvočísla.

Multiplikativní inverze

Definice

Pro $a \in \mathbb{Z}$ a $n \in \mathbb{N}$ je celé číslo x **multiplikativní inverzí**, pokud splňuje podmínku

$$a \cdot x \equiv 1 \pmod{n}. \tag{5}$$

Pro **nejmenší multiplikativní inverzi** platí, že x je nejmenší možnou kladnou multiplikativní inverzí k a a označujeme ji a^{-1}.

Z Malé Fermatovy věty přitom plyne, že

$$a^{-1} \equiv a^{p-2} \pmod{p}. \tag{6}$$

pro $a \in \mathbb{Z}$ a prvočíselná $p \in \mathbb{N}$ taková, že $p \nmid a$.

Multiplikativní inverze
Příklad

Výpočet inverze

Chceme spočítat a^{-1} pro $n = 11$ a $a = -3$. Volíme postupně $x = 1, 2, \ldots,$ první kladné číslo x splňující vztah (5) je $x = 7$: $-3 \cdot 7 \equiv 1 \pmod{11}$.

Výpočet inverze pomocí Malé Fermatovy věty

Použitím Malé Fermatovy věty (6) máme $a^{-1} \equiv (-3)^{11-2} \pmod{11}$, tedy $a^{-1} \equiv -19683 \pmod{11}$ což je to samé, jako $a^{-1} \equiv 7 \pmod{11}$ protože jde o stejnou třídu kongruence.

Zkuste si to nyní sami pro $n = 7$ a $a = 5$.

Tři námořníci, opice a kokosy

Problém

Na pustém ostrově ztroskotají tři námořníci. Jediná potrava, kterou během dne našli, je hromada kokosových ořechů.

V noci se první námořník probudí, spravedlivě rozdělí hromadu na tři díly, přičemž jeden kokos zbyde – ten dostane opice. Svou třetinu námořník ukryje, zbytek navrší zpátky a jde zase spát. Postupně hromadu stejným způsobem „třetina pro mne, jeden kokos opici, zbytek vrátit“ zmenší jeho oba druhové.

Ráno si hromadu rozdělí na třetiny, opět zbyde jeden kokos, ten dostane opice.

Kolik musí být v původní hromadě kokosů, aby to fungovalo?

Tři námořníci, opice a kokosy

Řešení (1)

První námořník začíná s hromadou obsahující $n \equiv 1 \pmod 3$ kokosových ořechů.

Druhý námořník dělil hromadu s

$$m_1 = \frac{2(n-1)}{3} \equiv 1 \pmod 3$$

ořechy, třetí námořník přerozděloval

$$m_2 = \frac{2(m_1-1)}{3} \equiv 1 \pmod 3$$

ořechů a ve zbylé hromadě jich muselo zůstat

$$m_3 = \frac{2(m_2-1)}{3} \equiv 1 \pmod 3.$$

Tři námořníci, opice a kokosy
Řešení (2)

Hodnotu m_3 spočteme jako

$$m_3 = \frac{2}{3}m_2 - \frac{2}{3} = \cdots = \frac{8}{27}n - \frac{38}{27} \equiv 1 \pmod 3$$

a rovnici v modulární aritmetice řešíme pro n

$$8n - 38 \equiv 27 \pmod{81} \Rightarrow 8n \equiv 65 \pmod{81}.$$

Dělit osmi nemůžeme, můžeme ale násobit multiplikativní inverzí (pro jejíž výpočet nelze použít Fermatovu větu – proč?):

$$n \equiv 8^{-1} \cdot 65 \equiv 71 \cdot 65 \equiv 79 \pmod{81}.$$

Nejmenší počet kokosů v hromadě je tedy 79 (ale může být i 160, 241, ...).

Obsah konzultace

ISBN

Neboli *International Standard Book Number*

Má ho každá kniha, identifikuje zemi či region původu, nakladatele a vydání. Existuje ve verzi ISBN-10 a ISBN-13. Na poslední pozici každého ISBN je kontrolní cifra.

Příklad výpočtu kontrolní cifry ISBN-10

Mějme ISBN 0-552-13105-9. Kontrolní cifra ISBN-10 se počítá v modulu 11, pro případ zbytku 10 se použije znak X.
Kontrolní součet je
$0 \cdot 10 + 5 \cdot 9 + 5 \cdot 8 + 2 \cdot 7 + 1 \cdot 6 + 3 \cdot 5 + 1 \cdot 4 + 0 \cdot 3 + 5 \cdot 2 + 9 \cdot 1 = 143 \bmod 11 = 9$. Uvedené ISBN je opravdu platné.

Což takhle 80-85609-70-3?

Rodné číslo

Varianta po roce 1954

Jednoznačný identifikátor občanů ČR a SR obsahující údaj o datumu narození, pohlaví a do roku 2004 i lokalitě porodnice.

Příklad výpočtu kontrolní cifry

Muž narozen 22. února 1959, rozlišující trojčíslí 177 (Zlín?). Poslední cifra rodného čísla zajišťuje dělitelnost ciferného součtu jedenácti, musí mít proto hodnotu $590222177 \bmod 11 = 6$. Odpovídající rodné číslo má tvar 590222/1776.

O kohopak asi jde?

Generátory pseudonáhodných čísel

Matematické přiblížení k $U(0,1)$

Jednou z možností je **lineární kongruentní generátor** *(LCG, Linear Congruence Generator)*.

Jak funguje LCG

Uživatel zvolí x_0 (pevné nebo třeba odvozené od aktuálního času). Potom $x_{k+1} = (a \cdot x_k + b) \bmod m$, kde a, b a m jsou zvolené parametry určující kvality generátoru.
Jedna z možných voleb je třeba $a = 1664525$, $b = 1013904223$ a $m = 2^{32}$.

LCG jsou velmi citlivé na volby parametrů. Pokud dodržíme jisté předpoklady, generátor pracuje s periodou m, ale i to je v mnoha případech statistických výpočtů (například u vícerozměrné Monte Carlo integrace) žalostně málo.

Aritmetika velkých čísel

Co s čísly, která počítač nedokáže reprezentovat?

Registry v dnešních procesorech jsou většinou 32 nebo 64 bitové:

- největší binární číslo, s nímž počítač dokáže *pohodlně* pracovat, je tedy 2^{32} respektive 2^{64},
- největší binární číslo, jež můžeme reprezentovat v 1GB operační paměti, je $2^{1099511627776}$ ... jak rychle s ním ale budeme schopni počítat?

Jak se ale algoritmy typu RSA efektivně vypořádávají se sčítáním či násobením celých čísel v aritmetice velkých modulů (třeba 340282366920938463463374607431768211507)? Jak provádět operace s třídami čísel, která se do paměti počítače prostě nevejdou?

V příštím díle uvidíte

V příští přednášce si vysvětlíme Čínskou větu o zbytcích *(Chinese Reminder Theorem)*.

A podíváme se, jak funguje šifrovací algoritmus RSA.