ZÁPADOČESKÁ UNIVERZITA V PLZNI
FAKULTA PEDAGOGICKÁ
KATEDRA MATEMATIKY, FYZIKY A TECHNICKÉ VÝCHOVY

PRVOČÍSLA: KLASICKÉ A NOVÉ POZNATKY

BAKALÁŘSKÁ PRÁCE

Michaela Cajthamlová
Matematika pro vzdělávání

Vedoucí práce: doc. RNDr. Jaroslav Hora, CSc.

Plzeň 2024

Prohlašuji, že jsem bakalářskou práci vypracovala samostatně s použitím uvedené literatury a zdrojů informací.

V Plzni dne ……

.....

vlastnoruční podpis

Poděkování

Tímto bych ráda poděkovala doc. RNDr. Jaroslavu Horovi, CSc. za vedení mé bakalářské práce, za cenné rady, vstřícnost při konzultacích a za ochotu, kterou mi věnoval v průběhu zpracovávání práce.

Obsah

Úvod

Tato bakalářská práce je zaměřena na klasické a nové poznatky o prvočíslech. V práci je uvedeno, co to jsou prvočísla, jak je lze ověřovat či hledat. Také si ukážeme programy, které nám v tom mohou pomoci. Budeme se tedy pohybovat v odvětví matematiky zvaném teorie čísel. Často, například u prvočíselných testů a faktorizačních metod, se budeme pohybovat v modulární aritmetice.

V první kapitole se seznámíme s tím, co to prvočísla jsou a k čemu se používají. Také si zkusíme objasnit pojem velká prvočísla.

Protože se jedná o klasické a nové poznatky, začneme historií prvočísel. Zde je například uvedeno, že první zmínky o prvočíslech datujeme již do roku 6500 př.n.l. Také jsou zde zmíněni matematici, kteří jsou spojeni s prvočísly nebo se je snažili objevovat.

Třetí kapitola propojuje matematiku s informatikou, zde si ukážeme, jak používáme prvočísla při předávání klíčů a při šifrování RSA.

Nejrozsáhlejší částí bakalářské práce je čtvrtá a šestá kapitola. Ve čtvrté se budeme věnovat metodám faktorizace přirozených čísel. Přes ty nejsnadnější metody, které zvládají i děti na základní škole, se dostaneme k těm pokročilejším a náročnějším, zároveň i efektivnějším. Podobným způsobem je zpracovaná i šestá kapitola věnovaná testům prvočíselnosti. Každý test je krátce popsán, pak je zde uvedeno, jak test funguje a ke každému je vyřešen příklad.

Pátá kapitola obsahuje prvočíselnou funkci, díky které můžeme určit počet prvočísel mezi dvěma hodnotami.

Sedmá kapitola uvádí způsoby, jakými můžeme hledat velká prvočísla a představíme si jednu z nejznámějších stránek pro hledání prvočísel Mersennova typu. Také si ukážeme, jak pomocí Pocklinghtonova kritéria vyrobit prvočíslo s daným počtem cifer.

Osmá kapitola krátce představí nejrůznější polynomy, které nám generují prvočísla alespoň pro nějakou množinu. Tuto kapitolu uzavřeme mnohočleny s dvaceti šesti proměnnými.

Poslední kapitola je věnována ukázce matematických programů, které se dají používat při hledání prvočísel či ověřování prvočíselnosti.

1 Proč potřebujeme velká prvočísla?

Prvočísla p jsou přirozená čísla, která jsou dělitelná pouze dvěma čísly – jedničkou a sama sebou. Číslo 1 není prvočíslem, protože nemá dva různé dělitele. Přirozená čísla, která nejsou prvočísla a nejsou číslo jedna, jsou čísla složená. Základní věta aritmetiky nám říká, že každé přirozené číslo větší než 1 lze rozložit na součin prvočísel.

A jaké je tedy jejich použití? Slouží nám vůbec prvočísla k něčemu?

„Je však zřejmé, že hledání stále větších a větších prvočísel bylo pro matematiky od dávnověku intelektuální výzvou. Nemá smysl se ptát, k čemu bylo toto hledání dobré. K čemu je ‚dobré" malování obrazů, hraní šachů či zdolávání velehor? Konáme mnoho činností jen proto, že jsme lidé obdařeni intelektem a emocemi, kteří cítí – alespoň někteří z nás – vnitřní potřebu poznávat nepoznané. Zdolávat nezdolané a sdělovat jiným své vidění světa, své myšlenky a obohacovat se navzájem. Každý z nás, kdo svůj život více či méně spojil s matematikou, proto dobře rozumí i v této oblasti pohnutkám našich předchůdců." [1]

K čemu jsou nám v této době ale důležitá? Jejich použití najdeme nejvíce v informatice a matematice. V informatice se používají velká prvočísla hlavně pro šifrovaní zpráv, na elektronické podpisy a u generování náhodných čísel.

Co je to vlastně to velké prvočíslo? Každý na tuto otázku máme trochu jinou odpověď, někomu mohou připadat prvočísla „velká", pokud mají alespoň 100 cifer, pro někoho jsou to ta největší nalezená (miliony cifer). Pokud budou zmíněna velká prvočísla, jedná se o prvočísla p, q tak velká, že se jejich součin pq nedá snadno faktorizovat, tedy tak velká, aby se zpráva při jejich využití k šifrování nedala napadat někým zvenčí. Největší známé prvočíslo má 24 862 048 cifer. Takto velká čísla nejsou příliš používána. Nejčastěji se budeme pohybovat mezi stovkami až tisíci cifer.

2 Historie

Mezi nejznámější badatele na poli prvočísel patří Euklides či Fermat. Ovšem nebyli první, kteří si všimli výjimečnosti prvočísel. Za nejstarší náznak týkající se prvočísel považujeme nalezenou kost, která se našla v roce 1960 v rovníkové Africe. Kost Ishanga se datuje k roku 6500 př. n. l. a nyní je uložena v Královském muzeu přírodních věd v Bruselu.

Proč je ale tak zajímává? Do kosti jsou vyryté vrypy do tří sloupců po 4 skupinách. A právě v prvním sloupci najdeme čísla 11, 13, 17 a 19, což jsou prvočísla od deseti do dvaceti. Zda opravdu šlo o zkoumání prvočísel nebo pouze o náhodně napsaná čísla se již nikdy nedozvíme.

Dále si prvočísel všimli také Číňané a okolo roku 1000 př.n.l. přišli na zvláštnost při skládání fazolí do obdélníku. Když budeme skládat obdélník z 15 fazolí, budeme mít tři řady po pěti fazolích a žádný problém nenajdeme. Pokud se to ale budeme snažit poskládat ze sedmnácti fazolí, existuje jen jeden způsob, a to poskládat fazole do řady.

O prvočíslech se více dozvídáme až ve 4.st.př.n.l., kdy Euklides vydal svoji knihu Základy, ve které uvádí teorii prvočísel. V knize uvádí jeden velice důležitý poznatek, a to ten, že je prvočísel nekonečně mnoho.

Dalším objevem byla tabulka na hledání prvočísel mezi jedničkou a tisícem. Za jejím vytvořením stojí knihovník z Alexandrie, Erasthothenes. Této tabulce se budeme později věnovat na str. 28.

Na nějaké další novinky si pak prvočísla musela počkat do 17. století, kdy se o ně začal zajímat francouzský matematik Pierre de Fermat. Prohlásil, že $2^{2^N} + 1$ budou pro jakékoliv n prvočísla. Pro n = 1, 2, 3, 4 problém nenajdeme, pokud ale zkusíme n = 5, dostaneme číslo 4 294 967 297, které je dělitelné číslem 641. Na Fermata se ovšem nemůžeme vůbec zlobit, nebylo v jeho možnostech testovat dělitelnost čísla, které mělo deset cifer. Také odhalil, že prvočísla, která při dělení čtyřmi dávají zbytek jedna lze napsat ve tvaru součtu dvou kvadrátů. Fermat dokonce tvrdil, že má i důkaz, ale nikdy ho nezapsal.

Pro příklad si zkusme rozložit prvočíslo 29, velice snadno zjistíme, že by bylo součtem kvadrátů čísel 5 a 2, tedy $29 = 5^2 + 2^2$.

O svých matematických objevech si často dopisoval s francouzským mnichem Marinem Mersennem. Tento milovník hudby a stvořitel teorie harmonických tónů se také velice intenzivně zajímal o čísla, a proto si s Fermatem často vyměňovali své nápady. Na začátku si myslel, že $2^n - 1$ budou vždy prvočísla, později si však uvědomil, že to nebude platit vždy. A tak se snažil přijít na to, kdy bude toto tvrzení pravdivé. Nejprve zjistil, že pokud n nebude prvočíslo, nemůže ani výsledek být prvočíslem. Jeho konečné tvrzení bylo, že pro $n = 2, 3, 5, 7, 13, 17, 19, 31, 67, 127, 257$ bude výsledkem prvočíslo. Jak na to přišel, asi není nikomu známé, už jen kvůli faktu, že číslo 2^{257} má 77 číslic. Prvočíslům, která jsou vytvořena pomocí vzorce $2^n - 1$, říkáme Mersennova prvočísla.

Po Fermatovi a Mersennovi přichází další, pro nás velice známý matematik, Leonhard Euler. Eulerovi se podařilo aspoň z části dokázat některá Fermatova tvrzení. Byl prvním, kdo dokázal, že vzorec $2^{2^N} + 1$ přestane fungovat již pro $N = 5$. Vytvořil tabulku, která o mnoho přesahovala starší Erasthothenesovu tabulku, nalezl a zapsal prvočísla až do 100 000. Přišel, podobně jako Fermat nebo Mersenne, se vzorcem na hledání prvočísel $x^2 + x + q$, kde q může být rovno 2, 3, 5, 11 nebo 17 a x bude nejvýše $q - 2$. Snažil se také o nalezení vzorce, který by dokázal vypsat všechna prvočísla. Euler v roce 1751 napsal: „Existují záhady, které lidstvo nerozluští nikdy. Abychom se o tom přesvědčili, stačí se letmo podívat na soupis prvočísel a ihned pochopíme, že nejsou řízena žádným řádem ani pravidlem.“ [2]

Předposledním matematikem, o kterém se zmíníme, je Carl Friedrich Gauss. Již v patnácti letech zjistil spojitost mezi logaritmy a prvočísly. Oproti předchozím se nesnažil najít všechna prvočísla, ale snažil se přijít na počet prvočísel v rozmezí od 1 do N. Tuto funkci označil jako $\pi(\mathrm{N})$. Vzorcem bychom ji mohli zapsat jako $N/\log(N)$. Mohlo by se zdát, že to má něco společného s konstantou π, ale jak jistě vidíme ze vzorce, tak to není pravda. Dále se touto funkcí budeme ještě zabývat podrobněji na str.19. Jeho odhad později vylepšil Legendre. Jeho funkce vypadala takto: $\frac{N}{\log(N)-1{,}08366}$. Gauss ovšem nezůstal pozadu a zavedl funkci $Li(N)$. Nakonec byla mnohem přesnější než Legenderova.

Poslední matematik, kterého zmíníme, není nikdo jiný než samotný Bernhard Riemann. V mládí ho přijali na Gymnasium Johanneum. Bernhard byl velice plachý, a tak se těžce začleňoval mezi své spolužáky. Aby nemusel trávit čas mezi spolužáky, utíkal do náručí knih. Díky panu řediteli, Schmalfussovi, měl přístup do knihovny, ve které se nacházelo

mnoho matematických děl. Jednou z knih byla i *Teorie čísel* od Adriena-Marie Legendrea. Jak již víme, Legendre a Gauss zkoumali souvislost logaritmické funkce a funkce, která počítá prvočísla. Kniha Riemanna tak nadchla, že ji přečetl během pár dnů *(pozn. kniha má 859 stránek)*. Když gymnázium dostudoval, chtěl se zapsat na jednu z nových univerzit. Bohužel pro něj si ale jeho otec přál, aby vstoupil do duchovního světa, aby měl pravidelný příjem a mohl finančně vypomáhat sestrám. Vydal se tedy na univerzitu v Göttingenu, kde byla vyučována teologie. Univerzita se sice dříve věnovala spíše teologii, ale v době, kdy nastoupil, už zde převládaly přírodní vědy. A tak si po čase Riemann chtěl zapsat matematiku a fyziku, nejprve se zeptal svého otce a doufal v souhlas. Naštěstí mu to otec dovolil, a tak mohl Riemann začít studovat to, co chtěl. Avšak už po roce mu univerzita neměla co nabídnout. V roce 1847 se přestěhoval do Berlína, kde začal navštěvovat Berlínskou univerzitu. Na této škole se seznámil s učitelem matematiky Dirichletem. Později spolu často diskutovali o objevech a zajímavostech ze světa matematiky. Riemann o prvočíslech napsal jeden jediný článek, který měl pouhých 10 stránek. Cílem článku bylo potvrdit, že čím větší čísla budeme dosazovat do Gaussovy funkce, tím blíže budeme k pravému výsledku. Časem se povedlo i tuto myšlenku dokázat. V článku se skrývalo ještě jedno tvrzení, které bylo bez důkazu, tzv. Riemannova hypotéza. Na její důkaz je dokonce vypsaná odměna jeden milion dolarů. [2]

Od roku 1952 se hledají největší velká prvočísla pomocí počítačů. Rok 1970 položil základy dnešní kryptografie, které se později budeme také věnovat. Kapitolu o historii zakončíme prozatím největším nalezeným prvočíslem $2^{82\,589\,933} - 1$, toto prvočíslo má skoro 25 milionů číslic a bylo objeveno v roce 2018 a jak již víme, jedná se o Mersennovo prvočíslo.

3 Šifrování

S prvočísly se setkáme v šifrování, v numerické analýze, aplikované matematice a v dalších aplikovaných vědách. Šifrování neboli kryptografie vznikla už velice dávno. Za první důkazy považujeme destičky z Mezopotámie, které vznikly okolo roku 1500 př.n.l. V nejstarších šifrách se ještě nesetkáme s matematikou, šifry byly založené převážně na přehazování nebo zaměňování symbolů. Kryptografie už dnes neslouží pouze k ochraně zpráv během přenosu, ale také k zabezpečení elektronických plateb nebo také k elektronickým podpisům. [3] Šifrování je „všude kolem nás“, aniž bychom většinou věděli, jak vlastně funguje. A není se čemu divit, vzhledem k propojení s matematikou se jedná někdy o opravdu složité algoritmy. Vzpomeňme si například na Caesarovu šifru. Takovouto šifru by rozluštily i děti ve škole, spočívá pouze v posunu v abecedě o daný počet znaků. Oproti tomu RSA šifrování se provádí s velkými prvočísly, kde musíme umět umocňovat a používat modulární aritmetiku, což už nemusí být pro všechny tak snadné.

Šifrování můžeme rozdělit na dvě podkategorie – symetrické šifrování a šifrování asymetrické. Jejich největším rozdílem je utajování klíčů. U symetrických šifer se používá stejného klíče na zašifrování i dešifrování. Klíč nesmí být veřejně známý a musí být bezpečně doručen adresátovi. Oproti tomu asymetrické šifrování má klíče dva, jeden pro zašifrování zprávy a druhý pro dešifrování zprávy. Jeden klíč je veřejný a druhý je tajný. Pokud tedy známe veřejný klíč, je skoro nemožné zjistit tajný klíč. Symetrické šifrování je oproti asymetrickému rychlejší, a proto se často asymetrické používá k šifrování klíčů pro symetrické šifrování.

Mezi symetrické šifrování patří například algoritmus DES nebo AES. My se v tomto textu podíváme hlavně na asymetrické šifrování, ve kterém hrají hlavní roli prvočísla.

3.1 Diffie – Hellman výměna klíčů

Než si představíme RSA šifrování, krátce se seznámíme s výměnou klíčů v asymetrickém šifrování. Diffie a Hellman jsou autoři algoritmu pro výměnu klíčů využívající modulární matematiku a prvočísla. Představíme si problém, který spolu vyřešili. Mějme Alici a Boba. Ti si chtějí sdílet svoje tajné klíče, ale nemají jinou cestu než tu nezabezpečenou. Všechny informace, které si vymění, se dozví také nepřítel Eva. Jak si mohou vyměnit klíč, aby se ho Eva nedozvěděla? Než si představíme algoritmus, musíme si nejprve vysvětlit, co je to primitivní kořen modulo n. Primitivním kořenem modulo n rozumíme

takové číslo g, pro které platí, že pro každé celé číslo a nesoudělné s n existuje takové celé číslo k, pro které platí $g^k \equiv a \bmod(n)$. Algoritmus pro výměnu klíče bude následující.

Diffie-Hellman algoritmus pro výměnu klíčů (dva účastníci)

1) Alice a Bob se spolu domluví na:
   - velkém prvočísle *p*
   - a na *g*, které je primitivním kořenem modulo *p*

   ⇨ Hodnoty p a g jsou veřejně přístupné.
2) Alice si zvolí své **tajné** celé číslo *a*
3) Bob si zvolí své **tajné** celé číslo *b*
4) Alice a Bob si vypočítají *A* a *B*
   - Alice vypočítá: $A \equiv g^a \pmod p$
   - Bob vypočítá: $B \equiv g^b \pmod p$
5) Následně si vymění hodnoty *A* a *B*
6) Po výměně si dopočítají *A'* a *B'*
   - Alice spočítá: $A' \equiv B^a \pmod p$
   - Bob spočítá: $B' \equiv A^b \pmod p$
7) Obě vypočítané hodnoty jsou stejné, protože platí:

   $$A' \equiv B^a \equiv (g^b)^a \equiv g^{ab} \equiv (g^a)^b \equiv A^b \equiv B'$$

Příklad – Diffie-Hellman

1. Mějme p = 937 a g = 5. Alice si zvolila své tajné číslo 2 a Bob 6. Zahrajte si na Alici a Boba a zkuste zjistit jaký je jejich tajný klíč.

Jako první vypočteme $A \equiv g^a$ *a* $B \equiv g^b$.

$A \equiv 5^2 \bmod(937)$ $\qquad$ $B \equiv 5^6 \bmod(971)$

$A \equiv 25 \bmod (937)$ $\qquad$ $B \equiv 15625 \bmod(937)$

$\boldsymbol{A = 25}$ $\qquad$ $B \equiv 633 \bmod(937)$

$\boldsymbol{B = 633}$

Nyní zjistíme jejich tajný klíč.

$A' \equiv 633^2 \bmod (937)$ $\qquad$ $B' \equiv 25^6 \bmod(937)$

$A' = 590$ $B' = 590$

Tajný klíč je 590. Pokud bychom někde udělali ve výpočtu chybu, nedostali bychom se ke stejnému výsledku.

Algoritmus je možné rozšířit i pro více účastníků.

Přestože se zdá, že Eva nemá způsob, jak dešifrovat tajný klíč, existuje jiný způsob, jak se dostat mezi účastníky. Říká se tomu „Man in the middle“. Stačí aby Eva předstírala, že je jeden z účastníků (Bob nebo Alice) a domluvila se s oběma stranami na jejím klíči, přes který si všechny zprávy může sama dešifrovat. Tento problém se dá vyřešit digitálním podpisem, aby bylo jasné, že komunikujeme se správnou osobou. [4]

3.2 RSA šifrování

RSA šifrování patří mezi asymetrické šifrování, protože při šifrování využívá veřejný klíč. Jeho autory jsou Ron Rivest, Adi Shamir a Leonard Adleman. Prvním krokem RSA šifrování bude vytvoření klíčů. To probíhá pomocí následujícího algoritmu:

1) Zvolím dvě různá prvočísla p, q, kde $n = pq$
2) Spočítáme $\varphi(n) = (p-1)(q-1)$, kde $\varphi(n)$ je Eulerova funkce
3) Zvolíme si $e \in \mathbb{N}$ tak, aby $nsd[1](e, \varphi(n)) = 1$
4) Spočítáme $d \equiv e^{-1} (\mathrm{mod}(\varphi(n))$
5) Veřejný klíč $\mathrm{V}(n, e)$
6) Soukromý klíč $\mathrm{S}(n, d)$

Zašifrování zprávy proběhne následujícím způsobem:

1) Zvolíme si text, ten převedeme do číselné podoby $M < n$
2) Zašifrujeme text pomocí následujícího vzorce $C = M^e \mathrm{mod}\ n$

Posledním krokem bude dešifrování textu:

1) Máme zašifrovaný text C
2) Zprávu dešifrujeme pomocí vzorce $M = C^d \mathrm{mod}\ n$ [5]

Příklad – RSA šifrování

Nyní si zkusme zašifrovat a dešifrovat slovo PRVOCISLA, nejprve vytvoříme veřejný a soukromý klíč.

[1] nsd – největší společný dělitel

1. část *– vytvoření klíčů*

1) $p = 13, q = 19, n = 13 \times 19 = 247$
2) $\varphi(n) = (13 - 1)(19 - 1) = 12 \times 18 = 216$
3) $e = 7$
4) $d \times 7 \equiv 1 (\mathrm{mod}\ 216), d = 31$
5) *Veřejný klíč* $V(247{,}7)$
6) *Soukromý klíč* $S(247{,}31)$

2. část – *převod slova do číselné podoby*

Mějme tedy slovo PRVOCISLA.

1) *Zvolme si abecedu*[2]*, podle které budeme kódovat:* $A = 01, B = 02, C = 03, \ldots, I = 09, L = 12, O = 15, P = 16,\ R = 18,\ S = 19,\ V = 22, \ldots, Z = 26$
2) *M = PRVOCISLA = 16 18 22 15 03 09 19 12 01*
3) *Slova si rozdělíme vzhledem k délce tak, aby žádné M nebylo větší než 247*
   $M_1 = \ 161, M_2 = 82, M_3 = 215, M_4 = 030, M_5 = M_6 = 91, M_7 = 201$

3. část *– zašifrování slova (pomocí veřejného klíče)*

1) ${M_1}^e \ \mathrm{mod}(n) = \ 161^7 \ mod(247)$

   $161^1 \ \mathrm{mod}\ 247 = 161$

   $161^2 \ \mathrm{mod}\ 247 = 233$

   $161^4 \ \mathrm{mod}\ 247 = 196$

   $161^7 \ \mathrm{mod}\ 247$

   $= [(161^4 \ \mathrm{mod}\ 247) \times (161^2 \ \mathrm{mod}\ 247)$

   $\times (161^1 \ \mathrm{mod}\ 247)] \mathrm{mod}\ 247 = (196 \times 233 \times 161) \ \mathrm{mod}\ 247$

   $= 99$
2) $82^7 \mathrm{mod}\ 247 =$ *199*

   $82^1 \ \mathrm{mod}\ 247 = 82$

   $82^2 \ \mathrm{mod}\ 247 \ = \ 55$

   $82^4 \ \mathrm{mod}\ 247 \ = \ 61$
3) $215^7 \ \mathrm{mod}\ 247 \ = \ 123$
4) $30^7 \ \mathrm{mod}\ 247 \ = \ 30$

[2] V tomto případě jsme zvolili nejlépe pochopitelné zakódování, se kterým se v běžném kódování zpráv nesetkáme.

5) $91^7 \bmod 247 = 13$
6) $201^7 \bmod 247 = 163$
7) *Zašifrované slovo má tedy podobu* $C_1 = 99, C_2 = 199, C_3 = 123, C_4 = 30, C_5 = C_6 = 13, C_7 = 163$

4.část *– dešifrování slova (pomocí soukromého klíče)*

1) $M_1 = C_1^d \bmod n = 99^{31} \bmod 247$

   $99^1 \bmod 247 = 99$

   $99^2 \bmod 247 = 168$

   $99^4 \bmod 247 = 66$

   $99^8 \bmod 247 = 157$

   $99^{16} \bmod 247 = 196$

   $M_1 = (196 \times 157 \times 66 \times 168 \times 99) \bmod 247 = 161$
2) $M_2 = 199^{31} \bmod 247 = 82$
3) $M_3 = 123^{31} \bmod 247 = 215$
4) $M_4 = 30^{31} \bmod 247 = 30$
5) $M_5 = M_6 = 13^{31} \bmod 247 = 91$
6) $M_7 = 163^{31} \bmod 247 = 201$

Dostaneme tedy 161 82 215 30 91 91 201, rozdělíme si čísla na dvojice a dostaneme 16 18 22 15 30 91 91 20 1, můžeme si všimnout, že zde máme číslo 30, které ale v naší abecedě neexistuje, a tak doplníme před trojku nulu, kterou jsme při modulu vynechali. Výsledek bude nakonec vypadat takto 16 18 22 15 03 09 19 12 01, jak vidíme, po dekódování dostaneme naše slovo PRVOCISLA.

4 Faktorizace přirozených čísel

Úplnou faktorizací přirozených čísel míníme rozklad na součin prvočísel. S tím se děti setkávají již na základní škole, kde se ke zjištění prvočísel používají dvě metody, buď graficky pomocí stromu nebo pomocí tabulky.

4.1 Tabulková metoda

Mějme nějaké číslo n. Vytvoříme si tabulku, která bude mít dva sloupce, do prvního budeme psát dělitele a do pravého dělence. Prvočíselný rozklad dostaneme v levém sloupci tabulky. Nejsnáze to pochopíme přímo na příkladu.

Příklad – tabulková metoda

Mějme číslo 420.

420	
2	210
5	42
2	21
7	3
3	1

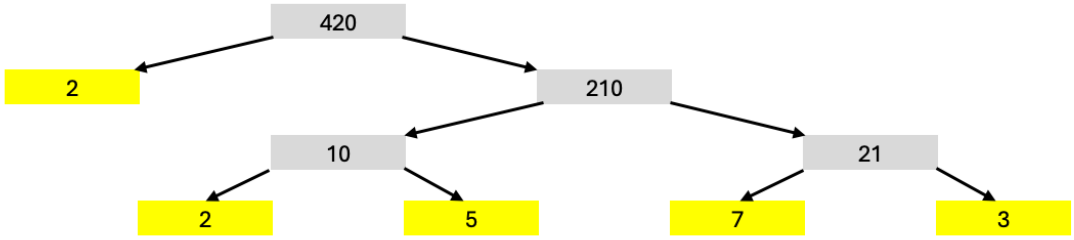
Tabulka 1: Faktorizace tabulkou

$$420 = 2 \times 5 \times 2 \times 7 \times 3$$

Prvočíselný rozklad čísla 420 bude $420 = 2 \cdot 2 \cdot 3 \cdot 5 \cdot 7$.

4.2 Grafická metoda – strom

Další velice snadná metoda. Nakreslíme si bublinku s číslem 420 kterou vždy rozdělíme na dvě větve, pokud půjdou bubliny dále něčím dělit, uděláme u nich opět vidličku, takhle pokračujeme, dokud nemáme všude prvočísla.



Obrázek 1: Grafická faktorizace

Rozšiřme si toto téma o nějaké další algoritmy, které nám pomůžou rozložit přirozené číslo na prvočísla.

4.3 Eulerova metoda

Touto metodou můžeme rozkládat pouze čísla, která se dají napsat dvěma různými způsoby jako součet čtverců.

$$N = a^2 + b^2 = c^2 + d^2$$

Eulerův algoritmus

1. Dostaneme číslo N a najdeme dva součty čtverců, kterým odpovídá
2. Vzorec si upravíme na $a^2 - c^2 = d^2 - b^2$, a to můžeme také zapsat jako $(a - c)(a + c) = (d - b)(d + b)$
3. Spočítáme si $k = nsd((a - c), (d - b))$ a $n = nsd((a + c), (b + d))$
4. Dále dopočítáme zbývající věci z následujících vzorců: (vybereme si dva nebo jeden z nich.)

$$(a - c) = kl$$

$$(d - b) = km$$

$$(a + c) = mn$$

$$(d + b) = ln$$

$$l(a + c) = m(d + b)$$

5. N pak spočítáme jako: $N = [(k/2)^2 + (n/2)^2] \times (m^2 + l^2)$ [6]

Příklad – Eulerův algoritmus

1. $N = 10121$

$$N = 10000 + 121 = 100^2 + 11^2 = 80^2 + 61^2$$

$$(100 - 80)(100 + 80) = (61 - 11)(61 + 11)$$

$$k = nsd(20,50), n = nsd(180,72)$$

$$k = 10, n = 36$$

$$(100 - 80) = kl \quad 20 = 10l \quad l = 2$$

$$(61 - 11) = km \quad 50 = 10m \quad m = 5$$

$$N = [(10/2)^2 + (36/2)^2] \times (5^2 + 2^2) = 349 \times 29$$

Prvočíselný rozklad čísla 10121 bude $10121 = 29 \cdot 349$.

4.4 Frematova metoda

O této metodě jsme se krátce zmínili již v historické části, pojďme se ale na toto téma podívat trochu hlouběji.

Mějme přirozené číslo n. Pokud je možné toto číslo zapsat ve tvaru $a^2 - b^2$, kde $a, b \in \mathbb{N}$, pak $n = uv$, kde $u = a + b$ a $v = a - b$. Abychom dostali netriviální rozklad na prvočísla musí platit, že $a - b > 1$. [7] Toto nevypadá příliš složitě a u některých čísel si s tím velice snadno poradíme, co ale s většími čísly, kde už to tak viditelné nebude? Ukažme si nyní algoritmus, který nám s tímto problémem pomůže.

Fermatův algoritmus

1. Mějme přirozené číslo n
2. Zjistíme jeho odmocninu (zaokrouhlenou nahoru?) $a_0 = \lceil \sqrt{n} \rceil$
3. Nyní budeme zkoušet následující odmocniny, dokud nedostaneme přirozené číslo

$$\sqrt{a_0^2 - n} = b, \sqrt{(a_0 + 1)^2 - n} = b\ , \sqrt{(a_0 + 2)^2 - n} = b, \ldots$$

4. Pokud už máme přirozené číslo b, dopočítáme k němu číslo a, vezeme a_0 a přičteme k němu takové číslo, které jsme připočetli i pod odmocninou abychom dostali přirozené číslo.

5. Nyní už jen vypočteme $n = (a + b)(a - b)$

Příklad – Fermatův algoritmus

1. ***Mějme číslo $n = 1769$, pomocí Fermatova algoritmu zjistěte jeho prvočíselný rozklad.***

$$a_0 = \lceil \sqrt{n} \rceil = \lceil \sqrt{1769} \rceil = 43$$

$$\sqrt{43^2 - 1769} = 8{,}944 \ldots$$

$$\sqrt{(43+1)^2 - 1769} = 12{,}922 \ldots$$

$$\sqrt{(43+2)^2 - 1769} = 16$$

Již ve třetím kroku jsme narazili na číslo $b = 16$, snadno dopočteme $a = 43 + 2 = 45$.

$$1769 = (a + b)(a - b) = (45 + 16)(45 - 16) = 61 \times 29$$

Prvočíselný rozklad čísla 1769 bude $1769 = 61 \cdot 29$.

4.5 Kraitchik metoda

Maurice Kraitchik byl belgický matematik. ANO! Oproti Fermatovi se Kraitchik snažil najít $u^2 - v^2 = kn$ tedy, nesnažil se najít taková u, v, jejichž rozdíl by vytvořil přesně n, ale taková, která nám dají nějaký jeho násobek, můžeme to také přepsat tak, že hledáme $u^2 \equiv v^2 \bmod n$. Můžeme dostat dva různé výsledky $u \equiv \pm v \bmod n$ nebo $u \not\equiv \pm v \bmod n$. První možnost nás příliš nezajímá, ovšem ta druhá bude klíčová. Pokud je n liché a dělitelné alespoň dvěma prvočísly, pak alespoň polovinu řešení najdeme právě v druhém řešení. [8]

Kraitchikův algoritmus

1. Mějme přirozené číslo n
2. Najdeme druhou mocninu, která je větší než n
3. Použijeme Fermatův algoritmus
4. Pokud stále nic nemůžeme najít, budeme pokračovat podle Kraitchiaka
5. Jako u si označíme výsledky $a_0^2 - n, (a_0 + 1)^2 - n, (a_0 + 2)^2 - n, \ldots$
6. Za v označíme výsledky pod odmocninami
7. Pokud mezi v, najdeme čísla snadno faktorizovatelná, pak tyto čísla vynásobíme mezi s sebou.

8. u dostaneme tak, že vynásobíme $(a0 + k)$ které odpovídá číslům, která jsme faktorizovali ve v
9. zjistíme, zda $u \equiv v \bmod n$, pokud není, pokračujeme dalším krokem
10. Pomocí Eulerovo algoritmu zjistíme $nsd(u - v, n)$

Příklad – Kraitchikův algoritmus

1. $n = 20437$

$$143^2 = 20449$$

$$\sqrt{143^2 - 20437} = 6{,}164 \ldots$$

$$\sqrt{(143+1)^2 - 20437} = 12{,}041 \ldots$$

$$\sqrt{(143+2)^2 - 20437} = 15{,}937 \ldots$$

$$\sqrt{(143+3)^2 - 20437} = 19{,}104 \ldots$$

$$\sqrt{(143+4)^2 - 20437} = 21{,}868 \ldots$$

Mohli bychom takto pokračovat dál a doufat, že nám brzy vyjde přirozené číslo, my ale zkusíme rychlejší variantu.

v= 12, 299, 588, 879, 1172

Zkusíme některá čísla rozložit.

$$12 = 2^2 \times 3$$

$$588 = 2^2 \times 3 \times 7^2$$

$$v = 2^2 \times 3 \times 7 = 84$$

$$u = 143 \times 145 = 20735$$

$$v = 84 \equiv 7056 \, mod \, 20437$$

$$u = 20735 \equiv 298 \, mod \, 20437$$

Protože $1008 \not\equiv 298 \, mod \, 20437$*, můžeme už pomocí Euklidova algoritmu najít* $nsd(298 - 84, 20437) = nsd(214, 20437)$.

Euklidův algoritmus:

$$20437 = 95 \times 214 + 107$$

$$214 = 2 \times 107 + 0$$

$$nsd(214, 20437) \; = \; 107$$

$$20437 \; = \; 107 \times 191$$

$$107 \; \times 191$$

Rozklad čísla 20437 bude $20437 = 107 \cdot 191$.

4.6 Pollard rho metoda

Jedná se o algoritmus, který spadá mezi pravděpodobnostní metody, kterým se říká Monte Carlo metody. Algoritmus byl publikován v roce 1975 britským matematikem J. M. Pollardem. Také publikoval metodu $\rho - 1$ nebo například SNFS (Special Number Field Sieve).

Zvolme si funkci $f(x) = x^2 + 1 \bmod p$, kde p je prvočíslo. My při rozkladu ovšem prvočíslo ještě neznáme, a tak si musíme poradit trochu jinak. Mějme tedy funkci $F(x_{i+1}) \; = \; x_i^2 \; + \; 1 \bmod n$, kde $i = 0, 1, 2, 3..$ Po určitém čase se zacyklíme, právě proto se této metodě říká ρ. Nezacyklená část tvoří „ocásek“ a zacyklená kruh. Abychom potom přišli na p, budeme potřebovat spočítat $nsd(x_{2i} - \; x_i, n)$

Pollard rho algoritmus

1. Mějme číslo n, u kterého chceme zjistit prvočíselný rozklad
2. Zvolme si libovolné přirozené číslo x_0
3. Počítejme $F(x_{i+1}) \; = \; x_i^2 \; + \; 1 \bmod n$, dokud se nezacyklíme
4. Najdeme netriviální $nsd(x_{2a} - x_b, n) = p_1$
5. Abychom zjistili p_2, stačí spočítat $n \div p_2 = p_1$

Příklad – Pollard rho algoritmus

1. Mějme číslo $n = 667$, zvolíme si např. $x_0 = 5$.

$$x_1 \; = \; 5^2 \; + \; 1 \bmod 667 \; = \; 26$$

$$x_2 = \; 26^2 + 1 \bmod 667 \; = \; 10$$

$$x_3 = 101, \; x_4 = 197, x_5 = 124, x_6 = 36, x_7 = 630, x_8 = 36, x_9 = 630$$

Zacyklená část je tedy od x_6, všechno před tím je „ocásek“.

$$nsd(x_2 - x_1, n) \; = \; nsd(10 - 26{,}667) \; = \; 1$$

$$nsd(x_4 - x_2, n) = nsd(197 - 10, 667) = 1$$

$$nsd(36 - 101, 667) = 1$$

$$nsd(36 - 197) = 23$$

$$667 = 23 \times p_2$$

$$667 = 23 \times 29$$

Prvočíselný rozklad čísla 667 bude 23 a 29.

2. ***Mějme číslo $\mathbf{n = 1457}$, zvolíme si např. $\mathbf{x_0 = 2}$.***

$$x_1 = 2^2 + 1 \bmod 1457 = 5$$

$$x_2 = 5^2 + 1 \bmod 1457 = 26$$

$$x_3 = 677,\ x_4 = 832, x_5 = 150, x_6 = 646, x_7 = 615, x_8 = 863, x_9 = 243,$$
$$x_{10} = 770, x_{11} = 1359, \ldots$$

Protože jsme se stále nezacyklili, může se stát, že s naším x_0 nemusíme najít žádný netriviální kořen. Pokud nejde o prvočíslo, můžeme problém vyřešit tím, že změníme právě x_0. Zvolme tedy jiné x_0, $x_0 = 20$.

$$x_1 = 20^2 + 1 \bmod 1457 = 401$$

$$x_2 = 401^2 + 1 \bmod 1457 = 532$$

$$x_3 = 367,\ x_4 = 646, x_5 = 615, x_6 = \mathbf{863}, x_7 = 243, x_8 = 770, x_9 = 1359,$$
$$x_{10} = \mathbf{863}, x_{11} = 243, \ldots$$

$$nsd(x2 - x1, n) = nsd(131,1457) = 1$$

$$nsd(x4 - x2, n) = nsd(114, 1457) = 1$$

$$nsd(x6 - x4, n)$$

Euklidův algoritmus:

$$1457 = 6 \times 217 + 155$$

$$217 = 1 \times 155 + 62$$

$$155 = 2 \times 62 + 31$$

$$62 = 2 \times 31 + 0$$

$$nsd(217, 1457) = 31$$

$$1457 : 31 = 47$$

$$1457 = 31 \times 47$$

Prvočíselný rozklad čísla 1457 bude tedy $1457 = 31 \cdot 47$.

4.7 Kvadratické síto

Kvadratické síto patří mezi jednu z nejrychlejších faktorizačních metod. Vzhledem k tomu, že nepatří mezi ty jednodušší na počítání, ukažme si rovnou algoritmus pro její lepší pochopení.

Kvadratické síto – algoritmus

1. Mějme zadané číslo n
2. Zjistíme $a_0 = \lceil \sqrt{n} \rceil$
3. Projdeme několik hodnot $F(T) = T^2 - n$, kde T je postupně a_0, $a_1 = a_0 + 1$, $a_2 = a_0 + 2, a_3 = a_0 + 3, \ldots$
4. Vybereme si B-hladké číslo (B-hladká čísla jsou taková čísla, jejichž všechny prvočíselní činitelé jsou menší než B)
5. Začneme prosívat naše výsledky $F(T)$:
    a. Začnu prvním číslem a vydělím ho postupně prvočísly do čísla B, pokud ho nedělí ani jedno, toto číslo si můžeme ze seznamu vyškrtnout
    b. Čísla, která ho dělí si k němu zapíšeme
    c. Takto projdeme všechna zbylá čísla
6. Po prosívání budeme hledat taková čísla, která když vynásobíme mezi sebou, dají nám u všech prvočíselných kořenů sudé mocniny, pokud žádné nenajdeme, zopakujeme krok 3 a přidáme další hodnoty od místa, kde jsme skončili
7. Nyní vezmeme všechna T, která náleží k našim vybraným číslům
8. Označme si $a = \prod T_i^2$ a $b = (\prod p_i)^2$
9. Uděláme následující kongruenci $T_i^2 \times T_j^2 \equiv (\prod p_i)^2 \mod n$
10. Vypočítáme $nsd(a - b, n)$ např. pomocí Euklidova algoritmu

Příklad – Kvadratické síto

1. $n = 8611$

$$a_0 = 93, a_1 = 94, a_2 = 95, a_3 = 96, \ldots$$

$$F(a_0) = 93^2 - 8611 = 38$$

$$38, 225, 414, 605, 798, 993, 1190, 1389, 1590, 1793, 1998, 2205, \ldots$$

Budeme hledat 23 - hladká čísla:

$$38 = 2 \times 19$$

$$225 = 5^2 \times 3^2$$

$$414 = 2 \times 3^2 \times 23$$

$$605 = 5 \times 11^2,$$

$$798 = 2 \times 3 \times 7 \times 19,$$

~~$993 = 3 \times 331$~~

$$1190 = 2 \times 5 \times 7 \times 17,$$

~~$1389 = 3 \times 463$~~

~~$1590 = 2 \times 5 \times 3 \times 53$~~

~~$1793 = 11 \times 163$~~

~~$1998 = 2 \times 3^3 \times 37$~~

$$2205 = 5 \times 3^2 \times 7^2$$

Nyní budeme hledat taková čísla, která nám po vynásobení prvočíselného rozkladu vytvoří sudé mocniny. Můžeme si všimnout, že čísla 17 a 23, se u našich čísel objevují vždy jen jednou, a tak i tato čísla můžeme vyškrtnout.

38	2					19
225		3^2	5^2			
605			5		11^2	
798	2	3		7		19
2205		3^2	5	7^2		

Tabulka 2: Kvadratické síto

Z tabulky snadno vyčteme, že se jedná o čísla 605 a 2205. Nyní si zjistíme, pro jaká a_n nám vyšla tato čísla $F(96) = 605$ a $F(104) = 2205$.

$$104^2 \times 96^2 \equiv (5 \times 11^2) \times (3^2 \times 5 \times 7^2) \bmod 8611$$

$$(1373)^2 \equiv (5 \times 11 \times 3 \times 7)^2 \bmod 8611$$

$$(1373)^2 \equiv (1155)^2 \bmod 8611$$

$$\text{nsd}(1373 - 1155, 8611) = \text{nsd}(218, 8611)$$

Euklidův algoritmus:

$$8611 = 39 \times 218 + 109$$

$$218 = 2 \times 109 + 0$$

$$\text{nsd}(218,8611) = 109$$

$$8611 \div 109 = 79$$

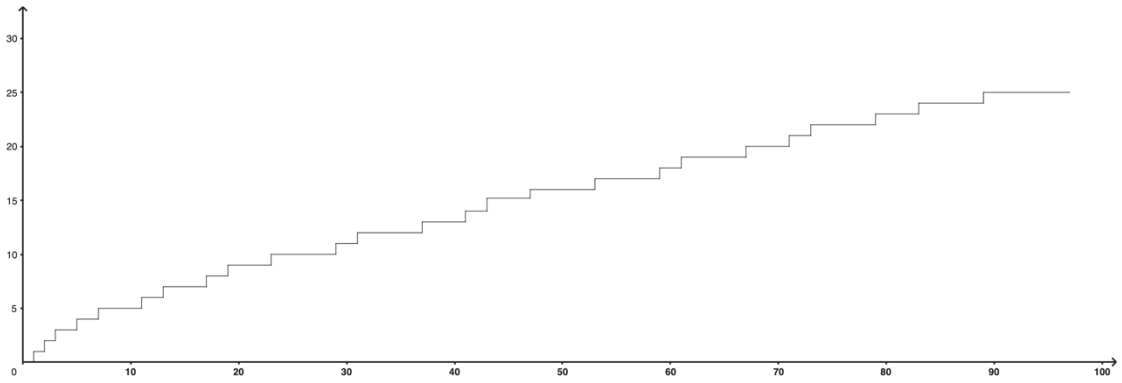
$$8611 = 79 \times 109$$

Prvočíselný rozklad čísla 8611 by byl $8611 = 79 \cdot 109$.

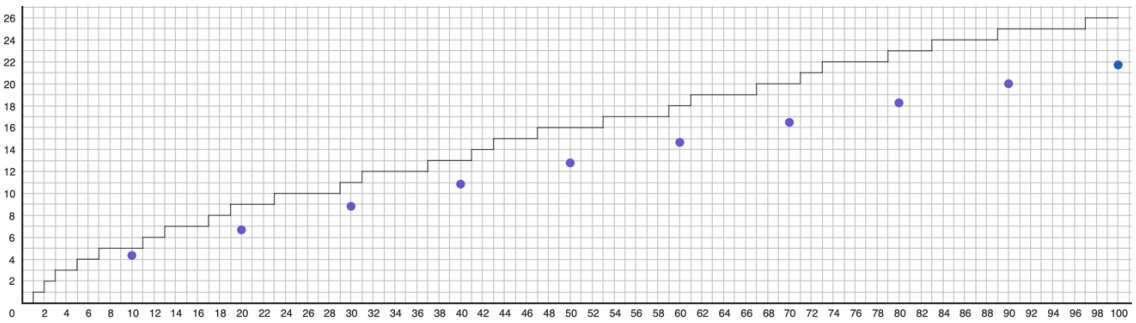
5 Funkce $\Pi(N)$

Funkci $\Pi(N)$ říkáme prvočíselná funkce, udává počet prvočísel do čísla N včetně.

O této funkci jsme se již zmínili v první kapitole věnované historii. S první myšlenkou hledat počet prvočísel mezi 1 a N čísly přišel Carl Friedrich Gauss. Gaussův odhad prvočísel stál na podobnosti s logaritmy. Další prvočíslo se bude nacházet přibližně jednou za $ln(N)$. K tomu abychom dostali počet prvočísel v nějakém rozmezí budeme potřebovat funkci $N/ln(N)$, kde $ln(N)$. Na obr.3 můžeme vidět jeho odhad od skutečnosti mezi čísly 1 a 100. Po objevení si zapsal vzorec na konec knihy, ovšem o svém nálezu nikomu nic nepověděl.



Obrázek 2: Počet prvočísel od 1 do 100

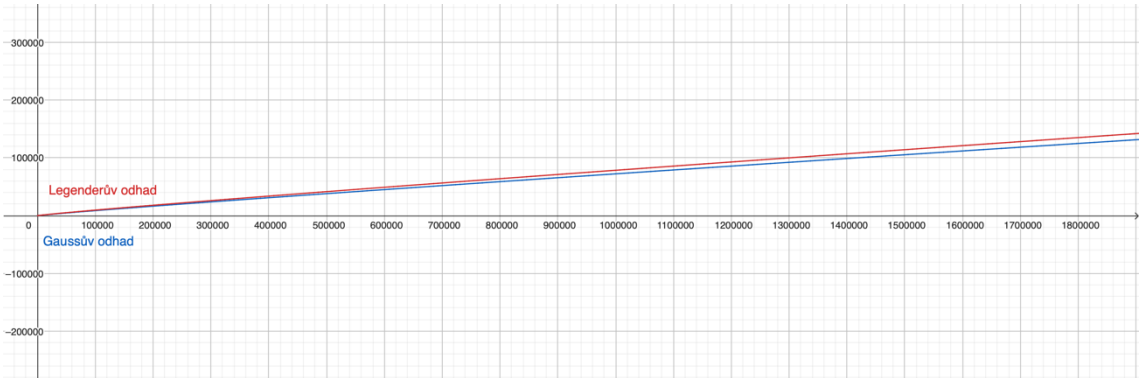


Obrázek 3: Gaussův odhad

Pokud bychom nyní vzali hodnoty do 10 000 000 pak bychom na grafu žádné schody už neviděli, viděli bychom pouze rostoucí „hladkou“ funkci.

O šest let později se matematik Adrien-Marie Legendre pochlubil se svým objevem. Legendreův odhad stál na následujícím vzorci $\frac{N}{\ln(N)-1{,}08366}$.

Legendre svůj vzorec pro výpočet počtu prvočísel ukázal světu dříve než Gauss, ne ale proto, že by na něj dříve přišel. Gauss nebyl příliš sdílný, když šlo o něco, co se nedalo dokázat, tak si několik let nechával vzorec sám pro sebe.



Obrázek 4: Gaussův odhad vs Legenderův odhad

Časem se ukázalo, že Legenderův vzorec je sice přesnější pro menší prvočísla, ovšem u velkých prvočísel převládá Gaussův odhad.

V té době tedy vypadal Legenderův odhad lépe, a tak se Gauss zamyslel a svůj výpočet zdokonalil. Podle Gausse je pravděpodobnost, že číslo bude prvočíslem $\frac{1}{ln(N)}$. Tedy počet prvočísel mezi N čísly spočítáme jako Li(N) = $\frac{1}{\ln(2)} + \frac{1}{\ln(3)} + \cdots + \frac{1}{\ln(N)}$. Tuto funkci nazval logaritmickým integrálem. Gauss, který měl v té době tabulky prvočísel až do čísla 3 000 000, si mohl snadno ověřit, že pro větší čísla jsou výsledky $Li(N)$ lepší než Legenderův odhad.

V dnešní době již víme, že funkce $\pi(N)$, ač se to nezdá, jednou přeroste funkci $Li(N)$.

6 Testy prvočíselnosti

Testy prvočíselnosti jsou algoritmy, které nám pomáhají ověřit, zda je naše číslo prvočíslem. Tak to je úžasné, říkáte si, ale není to tak růžové, jak se může zdát. Mnoho testů nám dokáže stoprocentně určit, zda se jedná o číslo složené nebo o prvočíslo. Existují ale i pravděpodobnostní testy, které pouze říkají, že zadané číslo je pravděpodobně prvočíslem.

6.1 Dělení

Prvočíslo je takové přirozené číslo, které je dělitelné jen samo sebou a číslem 1. Abychom tedy ověřili, že se jedná o prvočíslo, stačilo by dané číslo n vydělit všemi čísly, která jsou mezi čísly 1 a n. Pokud by všechna tato dělení vyšla se zbytkem, bylo by testované číslo n prvočíslem.

Příklad – dělení

1. $n = 23$

$$23 \bmod 2 = 1, \quad 23 \bmod 3 = 2, \quad 23 \bmod 4 = 3, \quad 23 \bmod 5 = 3,$$

$$23 \bmod 6 = 5, \quad 23 \bmod 7 = 2, \quad 23 \bmod 8 = 7, \ldots . 23 \bmod 22 = 1$$

Číslo 23 je tedy prvočíslem.

6.2 Eratosthenovo síto

Jedním z nejlehčích algoritmů pro ověření prvočíselnosti je Eratosthenovo síto. Jeho velikou nevýhodou je časová náročnost, tak s ním většinou budeme ověřovat malá prvočísla. Jeho podstata spočívá v tabulce, ve které postupně škrtáme násobky prvočísel. Ukažme si to na příkladu.

Budeme hledat prvočísla mezi čísly 1 a 100. Vytvoříme tabulku, ve které budeme mít všechna tato čísla napsaná.

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

Obrázek 5: Eratosthenovo síto 1

Jako první vyškrtneme číslo 1, protože se nejedná o prvočíslo. Nyní máme na řadě číslo 2. To je prvočíslem, a tak si ho zvýrazníme, projdeme zbytek tabulky a vyškrtneme všechna čísla dělitelná číslem 2.

	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~~

Obrázek 6: Eratosthenovo síto 2

Teď je na řadě číslo 3, protože ještě není vyškrtnuté, musí se jednat o prvočíslo, a tak ho zvýrazníme. Opět ze zbytku tabulky vyškrtneme všechny jeho násobky.

	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~~

Obrázek 7: Eratosthenovo síto 3

Tento postup opakujeme krok po kroku, dokud nedojdeme do konce tabulky.

	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~~

Obrázek 8: Eratosthenovo síto 4

Po projetí celé tabulky nám v ní zůstanou zvýrazněna pouze prvočísla.

6.3 Fermatův test prvočíselnosti

Mějme prvočíslo p a vzorec $a^p - a$. Pokud $p \mid a^p - a$, kde $1 \leq a < p$, jedná se o prvočíslo. Ukažme si nyní příklady na nějakých malých prvočíslech.

Příklad – Fermatův test

1. **Ověřte pomocí Fermatova testu, že číslo $p = 7$ je prvočíslem.**

$$2^7 - 2 = 128 - 2 = 126$$

$$3^7 - 3 = 2187 - 3 = 2184$$

$$4^7 - 4 = 16384 - 4 = 16380$$

$$5^7 - 5 = 78125 - 5 = 78120$$

$$6^7 - 6 = 279936 - 6 = 279930$$

Jak vidíme, všechna čísla jsou dělitelná číslem 7, jedná se tedy o prvočíslo.

2. **Ověřte pomocí Fermatova testu, že číslo $p = 18$ není prvočíslem.**

$$2^{18} - 2 = 262142$$
$$262142 \not\equiv 0 \pmod{18}$$

Číslo 18 není prvočíslem, k výsledku nám stačily pouze dva výsledky. Je jasné, že u takto malého čísla lze snadno poznat, zda je prvočíslem, nebo není.

U čísel větších můžeme mít problém s výpočtem, ne každý výpočetní program zvládne počítat vysoké mocniny čísla.

Pro číslo 7 jsme museli udělat pět kroků, pro číslo 2069 už bychom museli spočítat 2067 čísel a u všech ověřit, zda jsou dělitelná číslem 2069 a vzhledem k tomu, že se o prvočíslo jedná, opravdu by nám to zabralo spoustu času.

Představte si, že bychom kontrolovali největší známé prvočíslo, doba trvání tohoto algoritmu by byla nepředstavitelně velká, a proto již dnes máme lepší metody na ověřování prvočíselnosti, které jsou efektivnější a rychlejší.

6.4 Fermatův pravděpodobnostní test prvočíselnosti

Jedná se o pravděpodobnostní test, který je založen na malé Fermatově větě. Ta zní následovně: Pro $a \in \mathbb{Z}$ a prvočíslo $p \in \mathbb{N}$ takové, že $p \nmid a$ platí $a^{p-1} \equiv 1 \pmod{p}$ resp. $a^p \equiv a$.

Příklad – Fermatův test prvočíselnosti

1. ***Zjistěte pomocí Fermatova pravděpodobnostního testu, zda je číslo $p = 113$ prvočíslem.***

Nejprve zvolme číslo a.

$$a = 3$$

$$a^{p-1} \equiv 1 \pmod{p}$$

$$3^{112} \equiv 1 \pmod{113}$$

Toto tvrzení je pravdivé, číslo 113 je pravděpodobně prvočíslo.

2. ***Zjistěte pomocí Fermatova pravděpodobnostního testu, zda je číslo $p = 561$ prvočíslem, pokud $a = 2$.***

$$2^{560} \equiv 1 \pmod{561}$$

Toto tvrzení je pravdivé, číslo 561 by mělo být pravděpodobně prvočíslem.

Tento test má jeden nedostatek. Existují čísla složená, která tímto testem projdou a jsou vyhodnocena jako prvočísla. Těmto číslům říkáme čísla Carmichaelova. Jsou to čísla, která splňují kongruenci $a^{p-1} \equiv 1 \pmod{p}$. Na takové číslo jsme narazili ve druhém příkladu, číslo 561 můžeme rozložit na součin čísel 3, 11 a 17, není tedy prvočíslem, jak nám v testu vyšlo.

6.5 Solovay-Strassenův test prvočíselnosti

Za tímto prvočíselným testem stojí Robert M. Solovay a Volker Strassen. Tento pravděpodobnostní test byl vytvořen v roce 1977. Pokud budeme mít číslo n, které bude číslo složené, pak pro něj existuje číslo a z intervalu $\{1, \dots, n-1\}$, takové že $D(a, n) = 1$ a $a^{(n-1)/2} \not\equiv \left(\frac{a}{n}\right) \bmod n$. [9] Pokud pro n najdeme a, pro které bude platit $a^{(n-1)/2} \equiv$

$\left(\frac{a}{n}\right) \bmod n$, poté je číslo n pravděpodobně prvočíslem. Oproti Fermatovu testu, Solovay-Strassen test rozezná Carmichaelova čísla od prvočísel, ovšem stále se jedná o pravděpodobnostní test, takže i zde může nastat chyba a test by Carmichaelova čísla nerozeznal. V tomto testu se využívá Jacobiho symbolu $\left(\frac{a}{n}\right)$, který je zobecněním Legendreova symbolu.

Abychom tomuto pojmu porozuměli, musíme si nejprve objasnit pojmy kvadratický zbytek a kvadratický nezbytek. Číslo a nazveme kvadratickým zbytkem modulo n, jestliže čísla a a n jsou čísla nesoudělná a platí $a \equiv x^2 \pmod{n}$. Pokud číslo x neexistuje, řekneme, že číslo a je kvadratickým nezbytkem.

Pro Jacobiho symbol platí:

$$\left(\frac{a}{n}\right) = \begin{cases} 1 & \text{pokud } a \text{ je kvadratický zbytek} \\ -1 & \text{pokud } a \text{ je kvadratický nezbytek} \end{cases}$$

Příklad – Solovay-Strassenův test

1. Ověřte prvočíselnost pro $n = 97$.

$$a = 2$$

$$2^{(97-1)/2} \not\equiv \left(\frac{2}{97}\right) \pmod{97}$$

Čísla 2 a 97 jsou čísla nesoudělná.

$14^2 = 196 \equiv 2 \ (mod\ 97) \Longrightarrow$ *jedná se o kvadratický zbytek.*[3]

$$2^{48} \not\equiv 1 \pmod{97}$$

$$a = 3$$

$$3^{48} \not\equiv \left(\frac{3}{97}\right) \pmod{97}$$

$$3^{48} \not\equiv 1 \pmod{97}$$

$$a = 4$$

$$4^{48} \not\equiv \left(\frac{4}{97}\right) \pmod{97}$$

$$4^{48} \not\equiv 1 \pmod{97}$$

$$a = 5$$

[3] Hledání kvadratických zbytků je prováděno přes webovou stránku https://www.walter-fendt.de/html5/mcz/legendresymbol_cz.htm

$$5^{48} \equiv \left(\frac{5}{97}\right) \pmod{97}$$

$$5^{48} \equiv -1 \pmod{97}$$

Protože je číslo 5^{48} *kongruentní s* -1*, číslo 97 je pravděpodobně prvočíslo.*

6.6 Lucas – Lehmerův test prvočíselnosti

Lucas-Lehmerův test prvočíselnosti používáme pouze k hledání prvočísel mezi Mersennovými čísly. Největší známé prvočíslo je právě Mersennovo číslo, jedná se o $2^{82\,589\,933} - 1$. Tímto testem bychom pomocí počítače mohli ověřit, zda se opravdu jedná o prvočíslo.

M1	M2	M3	M4	M5	M6	M7	M8	M9	M10
1	3	7	15	31	63	127	255	511	1023

Tabulka 3: Mersennova čísla

Lucas-Lehmerův test – algoritmus

1. Mějme číslo M_p
2. Spočítáme si počet kroků S_{p-2}
3. $S_0 = 4 \bmod M_p$
4. $S_i = (S_{i-1} \bmod Mp)^2 - 2 \mod M_p$
5. S výpočty pokračujeme až do S_{p-2}včetně
6. Pokud $S_{p-2} \equiv 0 \bmod M_p$, pak se jedná právě o prvočíslo

Příklad – Lucas-Lehmerův algoritmus

1. Je číslo $M_9 = 511$ prvočíslo?

$$P = 9, Sp - 2 = S7$$

$$S_0 = 4 \equiv 4 \bmod 511$$

$$S_1 = 42 - 2 = 14 \bmod 511$$

$$S_2 = 142 - 2 = 194 \bmod 511$$

$$S_3 = 194^2 - 2 = 37634 \equiv 331 \bmod 511$$

$$S_4 = 331^2 - 2 \equiv 205 \bmod 511$$

$$S_5 \equiv 121 \bmod 511$$

$$S_6 \equiv 331 \bmod 511$$

$$S_7 \equiv 205 \bmod 511$$

$$205 \neq 0$$

Nejedná se tedy o prvočíslo, jeho prvočíselný rozklad by byl 7 a 73.

2. Je $M_{13} = 8191$ prvočíslem?

$$P = 13, SP - 2 = S11$$

$$S_0 = 4 \bmod 8191$$

$$S_1 = 14 \bmod 8191$$

$$S_2 = 194 \bmod 8191$$

$$S_3 = 4870 \bmod 8191$$

$$S_4 = 3953$$

$$S_5 = 5970$$

$$S_6 = 1857$$

$$S_7 = 36$$

$$S_8 = 1294$$

$$S_9 = 3470$$

$$S_{10} = 128$$

$$S_{11} = 0$$

Mersennovo číslo M_{13} je prvočíslo.

6.7 Pépinův test prvočíselnosti

Pépinův test používáme k ověření Fermatových čísel. Nejznámějším zatím ověřeným Fermatovým číslem je prvočíslo F_4.

F1	F2	F3	F4	F5	F6
5	17	257	65537	4294967297	1,84467E+19

Tabulka 4: Fermatova čísla

Pépin test – algoritmus

1. Mějme F_n (Fermatovo prvočíslo), tedy $F_n = 2^{2^x} + 1$, kde $n > 1$
2. Pokud platí $3^{(F_n-1)/2} \equiv -1 \pmod{F_n}$, pak se jedná o prvočíslo [10]

Příklad – Pépin test

1. $F_2 = 17$

$$3^{(17-1)/2} \equiv -1 \pmod{17}$$

$$3^8 \equiv -1 \pmod{17}$$

$$6561 \equiv -1 \pmod{17}$$

$$16 \equiv -1 \pmod{17}$$

A protože číslo 16 je kongruentní s -1*, číslo* 17 *je tedy prvočíslem.*

6.8 Miller-Rabinův test prvočíselnosti

Jedná se o pravděpodobnostní test. Pokud zadáme do tohoto testu číslo, můžeme zjistit, zda se jedná pravděpodobně o prvočíslo. Takovýmto testům říkáme Monte Carlo testy.

Miller-Rabinův test – algoritmus

1. Mějme zadané číslo n
2. Najdeme k a m pro která platí $n - 1 = 2^k \times m$
3. Vybereme si číslo a z intervalu $1 < a < n - 1$
4. Aby bylo číslo pravděpodobně prvočíslem musí platit $a^m \equiv 1 \pmod{n}$ nebo $a^{2^r m} \equiv -1 \pmod{n}$, pro některé $0 \leq r < k$

Příklad – Miller-Rabinův algoritmus

1. $\boldsymbol{n = 161}$

$$161 - 1 = 2^4 \times 10$$

$$k = 4, m = 10$$

$$1 < a < 161$$

Zvolme například $a = 8$

$$8^{10} \not\equiv 1 \pmod{161}$$

$$8^{2*10} \not\equiv 1 \pmod{161}$$

$$8^{3*10} \not\equiv 1 \pmod{161}$$

Číslo 161 by nemělo být prvočíslem. Pokud bychom chtěli zjistit jeho faktorizaci, použili bychom na to metody z minulé kapitoly.

2. $\boldsymbol{n = 83}$

$$83 - 1 = 2^1 \times 41$$

$$k = 1, m = 41$$

$$1 < a < 83$$

$$a = 2$$

$$2^{41} \not\equiv 1 \pmod{161}$$

$$2^{2*41} \equiv 1 \pmod{161}$$

Číslo 83 by mělo být prvočíslem.

6.9 AKS test prvočíselnosti

Tento algoritmus byl představen v roce 2002 autory Agrawal, Kayal a Saxena, po kterých je pojmenován. Jedná se o polynomiální test prvočíselnosti. Jedná se ovšem o algoritmus, který se v praxi nepoužívá, spíše se vyučuje na univerzitách.

Vzhledem ke složitosti tohoto algoritmu si ukážeme jeho hodně zjednodušenou podobu, která nám také ověří prvočíselnost.

AKS – zjednodušený algoritmus

1. Mějme číslo n, u kterého chceme ověřit prvočíselnost
2. Číslo dosadíme do následujícího vzorce $(x-1)^p - (x^p - 1)$
3. Po dosazení roznásobíme závorky
4. Pokud jsou čísla u členů polynomů dělitelná n, tak se jedná o prvočíslo

Příklad – AKS zjednodušený

1. $\boldsymbol{p = 3}$

$$(x-1)^3 - (x^3 - 1) = (x^3 - 2x^2 + x - x^2 + 2x - 1) - x^3 + 1 = -3x^2 + 3x$$

Z výsledku vidíme, že číslo 3 je dělitelné číslem 3 a tím pádem je číslo 3 prvočíslem.

2. $\boldsymbol{p = 4}$

$$(x-1)^4 - (x^4 - 1) = x^4 - 4x^3 + 6x^2 - 4x + 1 - x^4 + 1 = -4x^3 + 6x^2 - 4x + 2$$

My víme, že číslo 4 není prvočíslo, z výsledku by mělo vycházet to samé, a protože 6 ani 2 nejsou dělitelné 4, jedná se opravdu o číslo, které není prvočíslem.

Snadno si povšimnete spojitosti s binomickou větou.

7 Hledání velkých prvočísel

Hledání velkých prvočísel je potřeba zejména v kryptografii.

Do doby, než měli matematici k dispozici výkonné počítače, hledali prvočísla pomocí Eratosthenova síta anebo prostým dělením. Mohlo by se zdát, že například v 18. století mohli mít tak hotovou tabulku prvočísel do 10 000. Ovšem takový Gauss měl ve svých sedmdesáti letech tabulku hotovou až do čísla 3 000 000 a to je něco neuvěřitelného [11]. Často se nová prvočísla hledala jako „pokus-omyl“, občas se dokonce našel i někdo, kdo prohlásil, že dané číslo je prvočíslem (aniž by to měl ověřené) a svět to tak přijal a až později museli matematici dokázat, že to tak není.

Představme si nyní J. F. Kulika, který se narodil v Lvově v roce 1793. Tento ukrajinský matematik strávil několik let v České republice a také zde zemřel. Napsal mnooho tabulek zaměřených na prvočísla a dělitele přirozených čísel, například *Handbuch mathematischer Tafeln* (1824), *Divisores numerorum decies centena milia non excedentium* (1825), ***Magnus Canon divisorum pro omnibus numeris per 2,3 et 5 non divisibilibus et numerorum primorum interjacentium ad milies centena milia accuratius ad 100 330 201***. V posledním zmíněném díle najdeme nejmenší dělitele čísel od čísla 3 033 001 až do 100 330 201. [12]

K hledání prvočísel můžeme využívat Eratosthenovo síto, které jsme si ukázali v předchozí kapitole. Vzhledem k jeho velké časové náročnosti není úplně nejlepším způsobem pro hledání opravdu velikých prvočísel. Můžeme si trochu proces usnadnit a skončit u $\sqrt{x}$, kde x je poslední číslo v tabulce.

Dále se hledají velká prvočísla v oblasti největších nalezených prvočísel a poté se používají testy prvočíselnosti jako například Miller-Rabinův test.

7.1 Pocklinghtonovo kritérium

Věta – Pocklinghtonovo kritérium

Nechť p je liché číslo a k je číslo přirozené takové, že p nedělí k a $1 \leq k < 2(p+1)$. Mějme $N = 2kp + 1$, pak platí, že N je prvočíslo, pokud existuje přirozené číslo a, takové že

- $2 \leq a < N$,
- $a^{kp} \equiv -1 \pmod{N}$,

- $nsd(a^k + 1, N) = 1$. [13]

Pokud bychom hledali prvočísla s přesně x číslicemi, můžeme použít právě Pocklinghtonovo kritérium. Každé naše N, bychom si přejmenovali na p_i. A k němu bychom poté počítaly další k_i. Počet cifer se nám bude zdvojnásobovat, a tak bychom se dostali až na požadovaný počet cifer.

Příklad – Pocklinghtonovo kritérium

1. **Najděte prvočíslo konstrukcí počínající $p = 17$.**

$$1 \leq k < 2(17 + 1)$$

$$1 \leq k < 36$$

Zvolme si tedy k, např. $k = 31$

$$N = 2kp + 1 = 2 \times 31 \times 17 + 1 = 1055$$

Můžeme si všimnout, že číslo 1055 je dělitelné číslem 5. Nejedná se tedy o prvočíslo a nemá ani smysl hledat k němu číslo a, protože by neexistovalo. Zkusme zvolit jiné k.

$$k = 30$$

$$N = 2 \times 30 \times 17 + 1 = 1021$$

Pokud číslo 1021 zkontrolujeme pomocí programu, vyjde nám, že se jedná o prvočíslo. Výpočet k jeho nalezení nebyl příliš složitý, ani nezabral tolik času. Ovšem zkusme to zkontrolovat a najít a, které by mělo existovat.

$$2 \leq a < N$$

$$2 \leq a < 1021$$

$$a^{kp} \equiv -1 \pmod{N}$$

$$a^{510} \equiv -1 \pmod{1021}$$

Zkusme jít od nejmenšího možného $a, a = 2$

$$2^{510} \equiv -1 \ (mod\ 1021)$$

To je pravdivé tvrzení, zkontrolujme ještě druhé:

$$nsd(2^{510} + 1, 1021) = 1$$

Toto tvrzení je také pravdivé, tím pádem je číslo 1021 prvočíslem.

2. Najděte prvočíslo, které bude mít více jak patnáct číslic.

Zvolíme libovolné p_1 a k němu vybereme k_1, zvolíme počáteční čísla například pětimístná, aby nám stačila pouze dvě hledání k nalezení více jak patnáctimístného čísla.

$$p_1 = 10859$$

$$1 \leq k_1 < 2(10859 + 1)$$

$$1 \leq k_1 < 21720$$

$$k_1 = 21714$$

$$N = 2kp + 1 = 2 \times 21714 \times 10859 + 1 = 471584653$$

Vzhledem k velikosti čísla zde nebudeme hledat a. Zda se skutečně jedná o prvočíslo ověříme pomocí webové stránky alperton.com

Číslo 471584653 *je prvočíslem. Použijeme ho jako naše* p_2.

$$p_2 = 471584653$$

$$1 \leq k_2 < 2(471584653 + 1)$$

$$1 \leq k_2 < 943169308$$

$$k_2 = 943169271$$

$$N = 2kp + 1 = 2 \times 943169271 \times 471584653 + 1 = 889\,568\,306\,769\,595\,927$$

Číslo $889\,568\,306\,769\,595\,927$ *je prvočíslem. Nalezli jsme tedy osmnáctimístné prvočíslo.*

Hledání čísla k může na první pohled vypadat jednoduše, ale ne vždy tomu tak je. V intervalu $(943169271, 943169308)$ *nenalezneme ani jedno k, které by pak vytvořilo prvočíslo* N. *A objevili bychom tam i další intervaly, ve kterých bychom k nenašli.*

7.2 GIMPS

GIMPS neboli Great Internet Mersenne Prime Search je projekt, který pátrá po prvočíslech mezi Mersennovými prvočísly. GIMPS vznikl v roce 1996 a zakladatelem je George Woltman [14]. K ověřování se používá optimalizovaný Lucas-Lehmerův test. Do projektu se může zapojit kdokoliv, kdo má chuť hledat. První nalezené číslo $M_{1398269}$ se nalezlo hned v roce 1996. Posledním zatím nalezeným prvočíslem (únor 2024) je $M_{82589933,}$ které je zároveň největším nalezeným prvočíslem. Nalezl ho Patrik Laroche. Za nalezení největších prvočísel je nabízená peněžní odměna.

Na webových stránkách mersenne.org se můžete dočíst, jak se zapojit, jaká čísla již byla nalezena, která z nich byla ověřena a spoustu dalších zajímavých informací.

Obrázek 9: GIMPS mersenne.org

8 Polynomy s mnoha prvočíselnými hodnotami

Existují polynomy, do kterých po dosazení určitých čísel dostaneme prvočísla. Neexistuje ale žádný, který by pro všechna čísla generoval jen prvočísla.

Polynomy tvaru $f(x) = x^2 + x + p$

Již v roce 1772 přišel Euler s pozoruhodným polynomem, který bude generovat prvočísla. Pro všechna $x \epsilon \mathbb{Z}$ uvažujme hodnoty polynomu : $f(x) = x^2 + x + 41$. Prvočísla dostaneme pouze pro čísla $x = 0, 1, \dots, 39$.

Prvočísla, která dostaneme z tohoto polynomu budou následující:

$$f(x) = x^2 + x + 41, x\epsilon\{0,1, \dots, 39\}: 41, 43, 47, 53, 61, 71, 83,97, 113, 131, 151, 173,$$
$$197, 223, 251, 281, 313, 347, 383, 421, 461, 503, 547, 593, 641, 691, 743, 797, 853,$$
$$911, 971, 1033, 1097, 1163, 1231, 1301, 1373, 1447, 1523, 1601$$

Eulerův polynom máme tedy ve tvaru $f(x) = x^2 + x + p$, kde p je prvočíslo a $x = 0,1 \dots, p - 2$. Abychom z polynomu dostali pouze prvočísla musíme za p dosadit $\{2, 3, 5, 11, 17, 41\}$. Zkusme si opět vypsat, jaká prvočísla dostaneme u těchto nových polynomů.

$$f(x) = x^2 + x + 2, x\epsilon\{0\} : 2$$
$$f(x) = x^2 + x + 3, x\epsilon\{0,1\} : 3, 5$$
$$f(x) = x^2 + x + 5, x\epsilon\{0, 1, 2, 3\} : 5, 7, 11, 17$$
$$f(x) = x^2 + x + 11, x\epsilon\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\} : 11, 13, 17, 23, 31, 41, 53, 67, 83, 101$$

Za následujícím polynomem stojí Adrien-Marie Legendre.

$$f(x) = x^2 + x + 17, x\epsilon\{0, 1, \dots, 15\} : 17, 19, 23, 29, 37, 47, 59, 73, 89,$$
$$107, 127, 149, 173, 199, 227, 257$$

Nyní si ukážeme, proč to neplatí například pro číslo 13. Zkusme si vypsat opět čísla která z polynomu dostaneme.

$$f(x) = x^2 + x + 13, x\epsilon\{0, 1, \dots, 11\} : 13, 15, 19, 25, 33, 43, 55, 69, 85, 103, 123, 145$$

Pokud bychom zvolili nějaké vlastní $p > 41$, ve výsledných hodnotách bychom samozřejmě nějaká prvočísla dostali, ovšem nevycházela by nám jen a pouze prvočísla.

Polynomy tvaru $f(x) = x^2 + ax + p$

Dalším polynomem, který nám bude generovat pouze prvočísla pro určitou množinu čísel může být například $f(x) = x^2 + 3z + 43\ x\epsilon\{0, 1, \dots, 38\}$. [15]

$$f(x) = x^2 + 3z + 43\ x\epsilon\{0, 1, \dots, 38\} : 43, 47, 53, 61, 71, 83, 97, 113, 131,$$
$$151, 173, 197, 223, 251, 281, 313, 347, 383, 421, 461, 503, 547, 593, 641, 691, 743, 797,$$
$$853, 911, 971, 1033, 1097, 1163, 1231, 1301, 1373, 1447, 1523, 1601$$

Polynomy tvaru $f(x) = ax^2 + p$

Už jsme si ukázali polynom $f(x) = x^2 + x + 17, x \epsilon \{0, 1, \dots, 15\}$, dalším polynomem, který dává samá prvočísla, je polynom $f(x) = 2x^2 + 29, x \epsilon \{0, 1, \dots, 28\}$. Jejich objevitelem je Adrien-Marie Legendre. Ukažme si, že se opravdu jedná o prvočísla a vypočítejme si všech 29 hodnot. [15]

$$f(x) = 2x^2 + 29, x \epsilon \{0, 1, \dots, 28\}: 29, 31, 37, 47, 61, 79, 101, 127, 157, 191, 229, 271,$$
$$317, 367, 421, 479, 541, 607, 677, 751, 829, 911, 997, 1087, 1181, 1279, 1381, 1487,$$
$$1597$$

Zajímavostí je například polynom $f(x) = x^6 + 1091$, který dává prvočísla až pro čísla 3906, 4620, … Celkem je polynom ověřen pro 35 čísel, která můžeme najít v OEIS A066386. [16]

Polynomy tvaru $f(x) = ax^2 + bx + p$

Fung a Ruby polynom mají různé koeficienty u členů x^2 a x. Polynomy mají následující tvar $f(x) = 36x^2 - 810x + 2753, x \epsilon \{0, 1, \dots, 44\}$, druhý je ve tvaru $f(x) = 47x^2 - 1701x + 10181, x \epsilon \{0, 1, \dots, 42\}$. [15,16]

J. Borox polynom má tvar $f(x) = 6x^2 - 34x + 4903$.

Další polynomy bychom mohli najít například od autorů F. Gobbo, Speiser, Honaker. [16].

Polynomy vyššího řádu

Ukažme si příklady dvou polynomů od týmu Ivan Kazmenko a Vadim Trofimov. První polynom je ve tvaru $f(x) = 66x^3 - 3845x^2 + 60897x - 251831, x \epsilon \{0,1, \dots, 45\}$, druhý vypadá následovně $f(x) = 45x^4 - 3416x^3 + 96738x^2 - 1212769x + 5692031, x \epsilon \{0,1, \dots, 42\}$. [17]

Dalšími, kdo se zabýval hledáním polynomů, které by nám generovala prvočísla, jsou Jaroslaw Wroblewski a Jean-Charles Meyrignac, stojí za mnoha polynomy, a tak si ukažme opět alespoň dva:

$$f(x) = (x^6 - 126x^5 + 6217x^4 - 153066x^3 + 1987786x^2 - 13055316x + 34747236)/36, x \epsilon \{0,1,\dots,54\}$$

$$f(x) = (3\mathrm{x}^4 - 386\mathrm{x}^3 + 14301\mathrm{x}^2 - 191518\mathrm{x} + 738676)/4, \mathrm{x} \epsilon \{0,1,\dots,48\}$$ [17]

Samozřejmě dalších polynomů existuje celá řada, můžeme si je tedy vybírat dle potřeby.

Mnohočlen

Na mnohočlenu, který by dával opravdu jen prvočísla začal pracovat i Matijasevič, práci ale nedotáhl do konce. O pět let později, v roce 1976 byl představen mnohočlen, který má 26 proměnných a generuje všechna prvočísla. [18] Mnohočlen vypadá takto:

$$\begin{aligned}(K+2)\{1 - [WZ + H + J - Q]^2 - [(GK + 2G + K + 1)(H + J) + H - Z]^2 \\ - [2N + P + Q + Z - E]^2 - [16(K+2)(N+1)^2 + 1 - F^2]^2 \\ - [E^3 (E+2)(A+1)^2 + 1 - O^2]^2 - [(A^2 - 1)Y^2 + 1 - X^2]^2 \\ - [16R^2 Y^4 (A^2 - 1) + 1 - U^2]^2 \\ - \left[\left((A + U^2 (U^2 - A))^2 - 1\right)(N + 4DY)^2 + 1 - (X + CU)^2\right]^2 \\ - [N + L + V - Y]^2 - [(A^2 - 1)L^2 + 1 - M^2]^2 \\ - [AI + K + 1 - L - I]^2 \\ - [P + L(A - N - 1) + B(2AN + 2A - N^2 - 2N - 2) - M]^2 \\ - [Q + Y(A - P - 1) + S(2AP + 2A - P^2 - 2P - 2) - X]^2 \\ - [Z + PL(A - P) + T(2AP - P^2 - 1) - PM]^2\}\end{aligned}$$

Do daného mnohočlenu dosazujeme pouze přirozená čísla (vyjma nuly), pokud nám výsledek vyjde kladný, jedná se o prvočíslo. Vypadá to, že prvočísla už budeme hledat opravdu snadno. Má to ale jeden velký háček, pokud si vyzkoušíme do mnohočlenu dosazovat různá čísla, zjistíme, že dostat kladné číslo není tak snadné, jak se může zdát.

Podle jiného zdroje vypadá výraz podobně, ale liší se jedním členem (ten je v textu zvýrazněn tučně). Za tímto stojí čtveřice matematiků – Jame P. Jones, Daihachiro Sato, Hideo Wada a Douglas Wiens. [19]

$$
\begin{aligned}
(K+2)\{1-[WZ+H+J-Q]^2-[(GK+2G+K+1)(H+J)+H-Z]^2 \\
-\ [2N+P+Q+Z-E]^2 \\
-[16(\boldsymbol{K}+\mathbf{1})^{\mathbf{3}}(K+2)(N+1)^2+1-F^2]^2 \\
-[E^3\ (E+2)(A+1)^2+1-O^2\ ]^2-[(A^2-1)Y^2+1-X^2\ ]^2 \\
-[16R^2\ Y^4\ (A^2-1)+1-U^2\ ]^2 \\
-\left[\left(\left(A+U^2\ (U^2-A)\right)^2-1\right)(N+4DY)^2+1-(X+CU)^2\ \right]^2 \\
-[N+L+V-Y]^2-\ [(A^2-1)L^2+1-M^2\ ]^2 \\
-[AI+K+1-L-I]^2 \\
-[P+L(A-N-1)+B(2AN+2A-N^2-2N-2)-M]^2 \\
-[Q+Y(A-P-1)+S(2AP+2A-P^2-2P-2)-X]^2 \\
-[Z+PL(A-P)+T(2AP-P^2-1)-PM]^2\ \}
\end{aligned}
$$

Ani u tohoto mnohočlenu jsem s hledáním hodnot nebyla úspěšná. Zůstává tedy otázkou, jaká jsou ta správná čísla pro kladný výsledek.

9 Aplikace na práci s prvočísly

V této kapitole se podíváme na aplikace, či webové stránky, které nám pomohou s faktorizací, ověřováním prvočíselnosti a hledáním prvočísel.

Představíme si tu dvě webové stránky – WolframAlpha a Alperton, jejichž používání není zpoplatněno a budou pro většinu problémů dostačující. Jako poslední se podíváme na zpoplatněný program, který se ale hojně využívá, a tím je Mathematica. Další programy, které by se daly použít jsou například Maple, MATLAB anebo třeba programovací jazyk Python, který má knihovny, které by se pro prvočísla také daly použít.

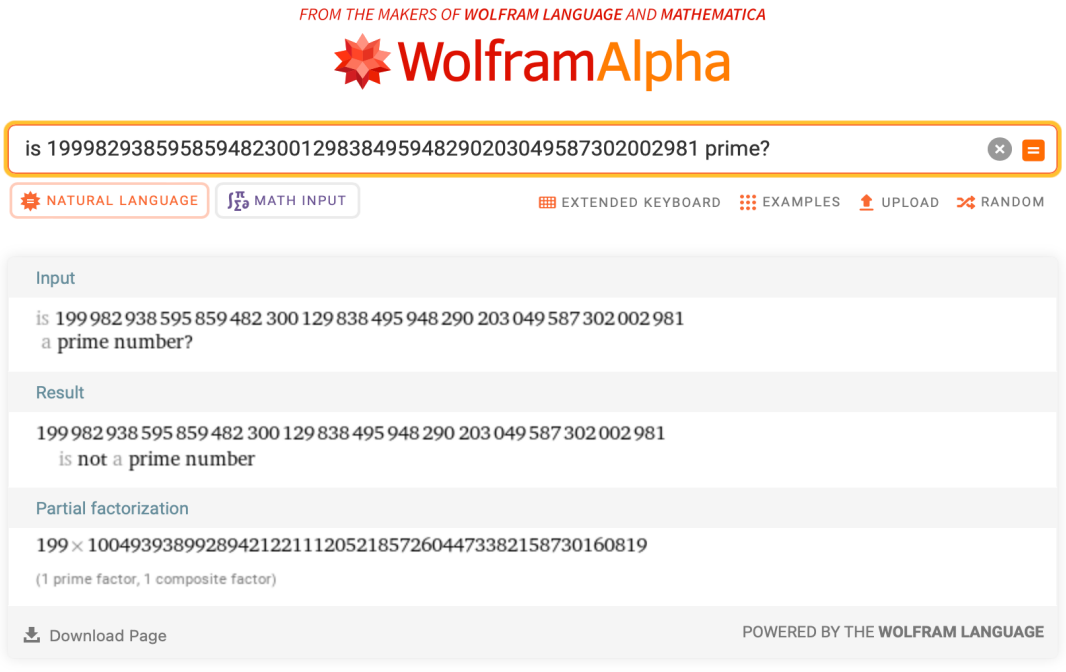
9.1 WolframAlpha

WolframAlpha je online nástroj, který je zdarma, takže ho můžeme používat ve školách i doma. Webová stránka nám dává odpovědi na naše dotazy. Neslouží pouze pro matematiku, dá se využít i pro vědecké a technické dotazy, kulturní, ale i každodenní dotazy ze života.

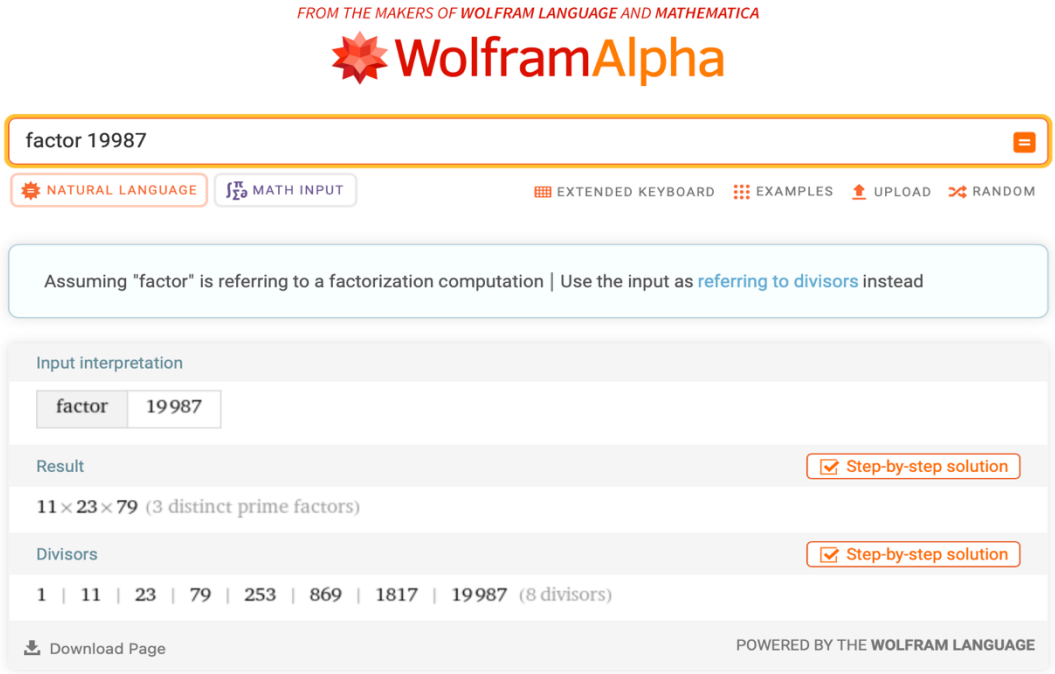
Pomocí tohoto programu můžeme zjistit, zda je číslo prvočíslem, můžeme také vypsat konkrétní prvočíslo v pořadí, generovat přesný počet prvočísel nebo najít nejbližší prvočíslo k určitému číslu. Také lze faktorizovat prvočísla, hledat prvočíselná dvojčata nebo hledat Mersennova a Fermatova prvočísla.

Kontrola prvočíselnosti, faktorizace čísel

Pokud bychom chtěli zjistit, zda je číslo *N* prvočíslo, stačí napsat do vyhledávače *„is N prime?“* V druhém políčku *Result,* pak zjistíme, zda se jedná o prvočíslo nebo ne. Pokud se nejedná o prvočíslo, program vypíše i prvočíselný rozklad. Pokud bychom chtěli vědět rovnou faktorizaci stačí zadat příkaz *„factor N*“.



Obrázek 10: WolframAlpha prvočíselnost

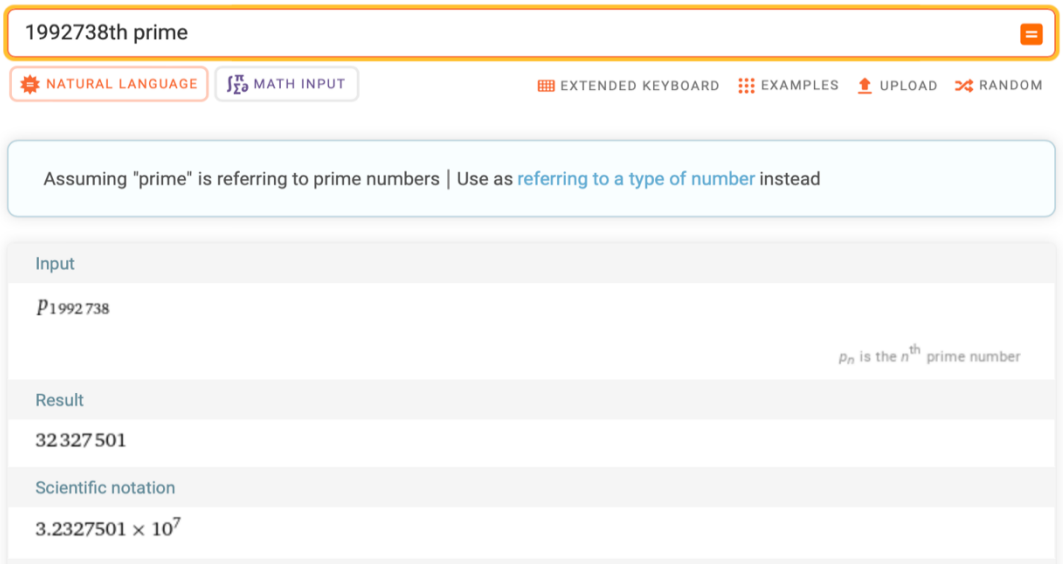


Obrázek 11: WolframAlpha faktorizace

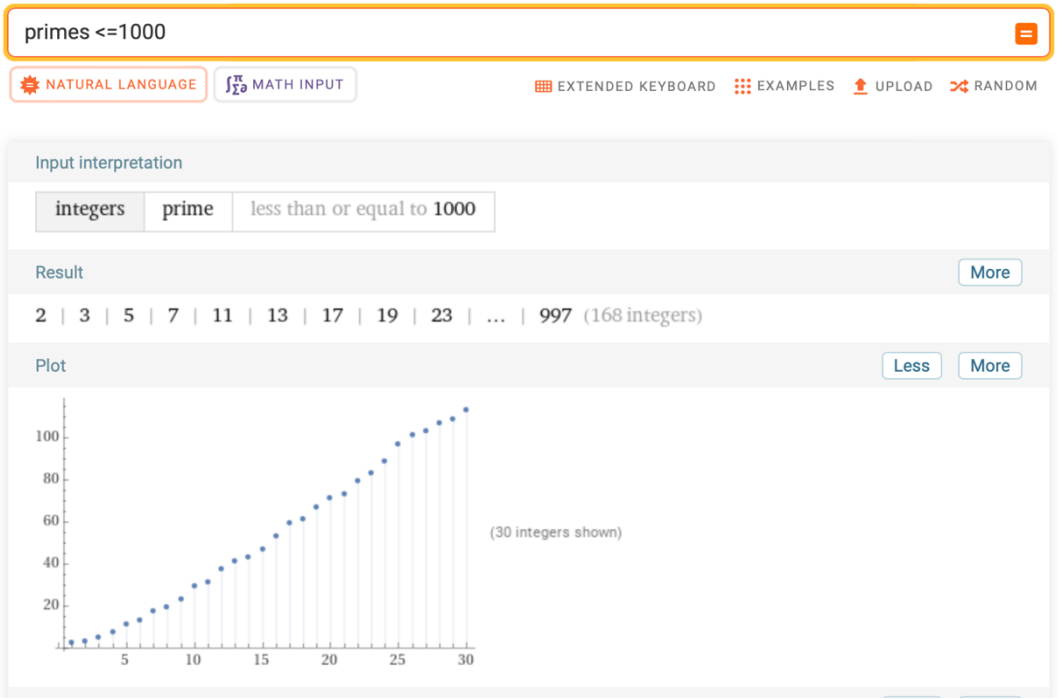
Generování prvočísel

Pokud bychom chtěli vypsat n-té prvočíslo stačí do příkazu zapsat *„Nth prime“*. Pokud bychom chtěli vypsat prvočísla v nějakém intervalu, máme dvě možnosti. První možností je, že budeme chtít všechna prvočísla do určitého čísla *„primes <=A“*. Druhou možností

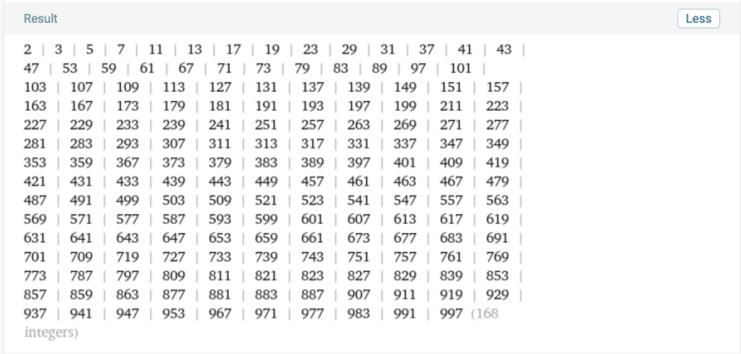
je pak výpis prvočísel mezi dvěma čísly *„primes between A and B“*. Pokud bychom neviděli všechny čísla jako na obr.12, klikneme u *Result* na tlačítko *More*.



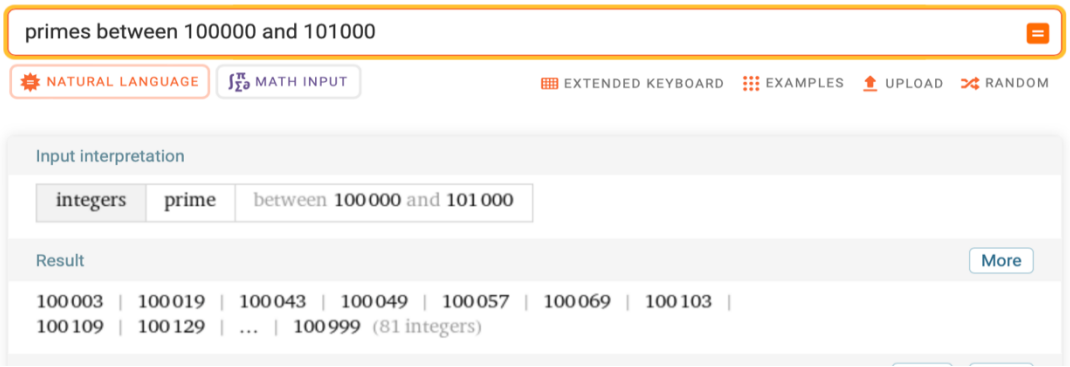
Obrázek 12: WolframAlpha Nté prvočíslo



Obrázek 13: WolframAlpha výpis prvočísel do N



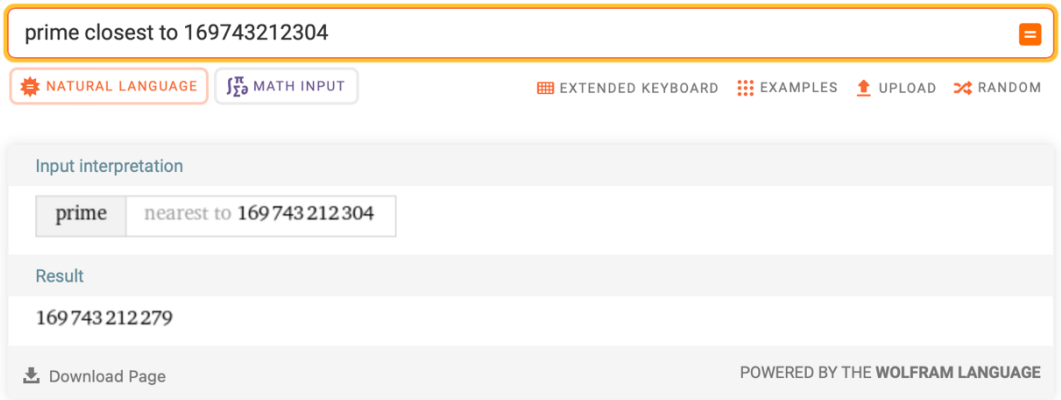
Obrázek 14: WolframAlpha tabulka prvočísel do N



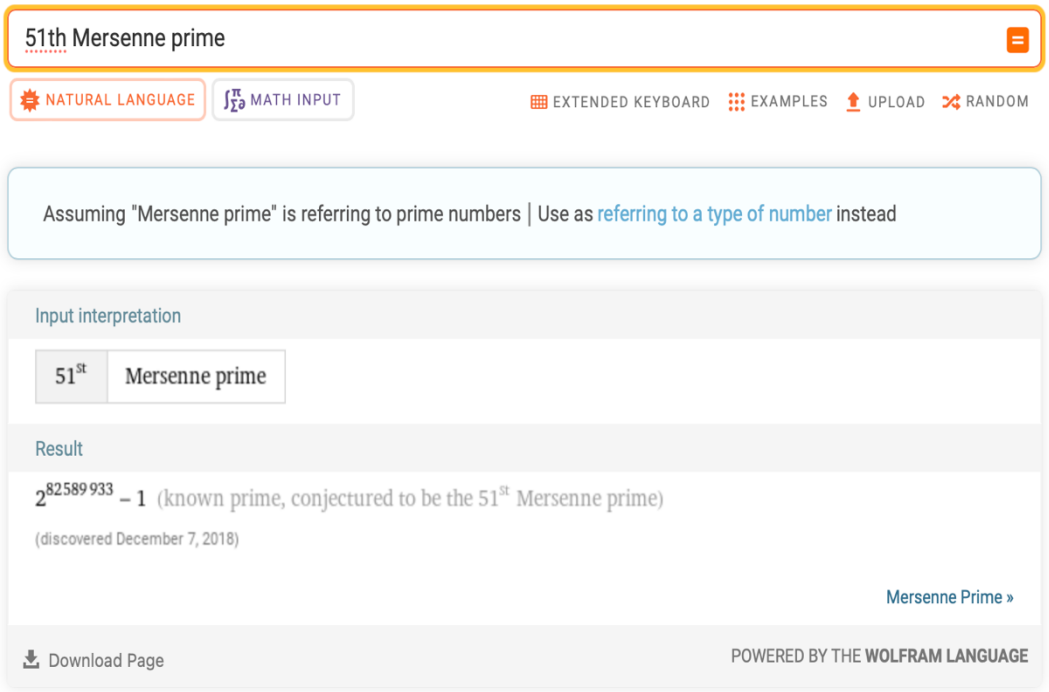
Obrázek 15: WolframAlpha prvočísla v intervalu od A do B

Hledání prvočísel

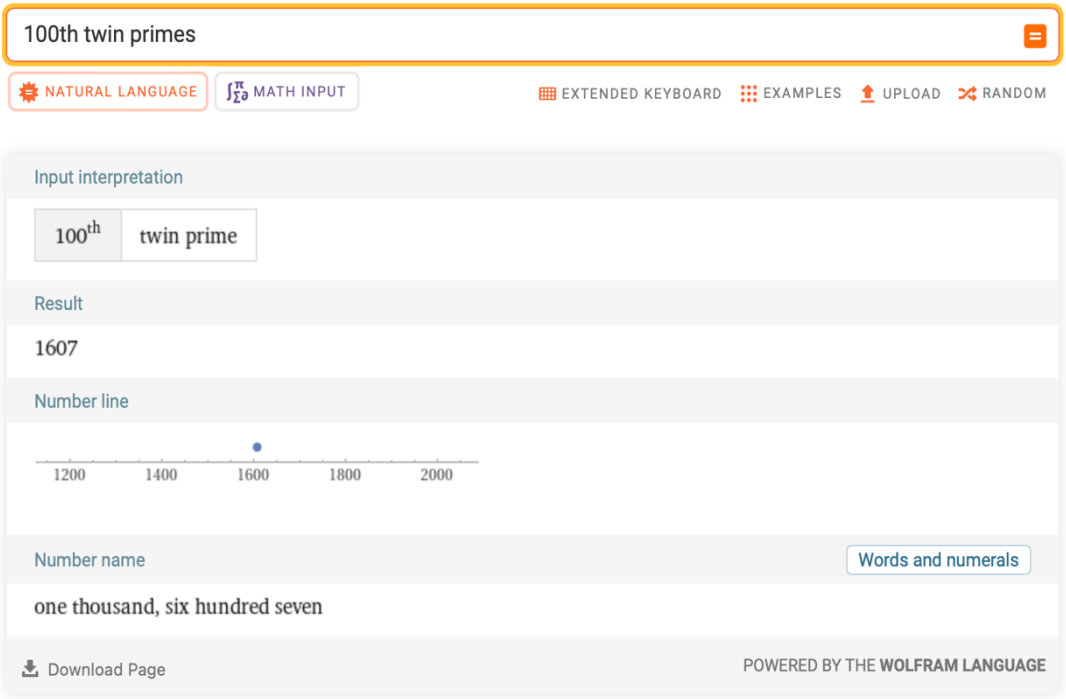
Můžeme hledat speciální prvočísla i prvočísla blízká určitému číslu. Například Mersennova prvočísla najdeme pomocí *„Nth Mersenne prime“*. Pro hledání prvočíselných dvojčat dáme do příkazu *„Nth twin prime“*. U této funkce vidím nevýhodu, že nám program nevypíše obě prvočísla najednou. Pokud bychom chtěli najít prvočíslo nejblíže k nějakému číslu, uděláme tak pomocí *„prime closest to N“*.



Obrázek 16: WolframAlpha nejbližší prvočíslo k N



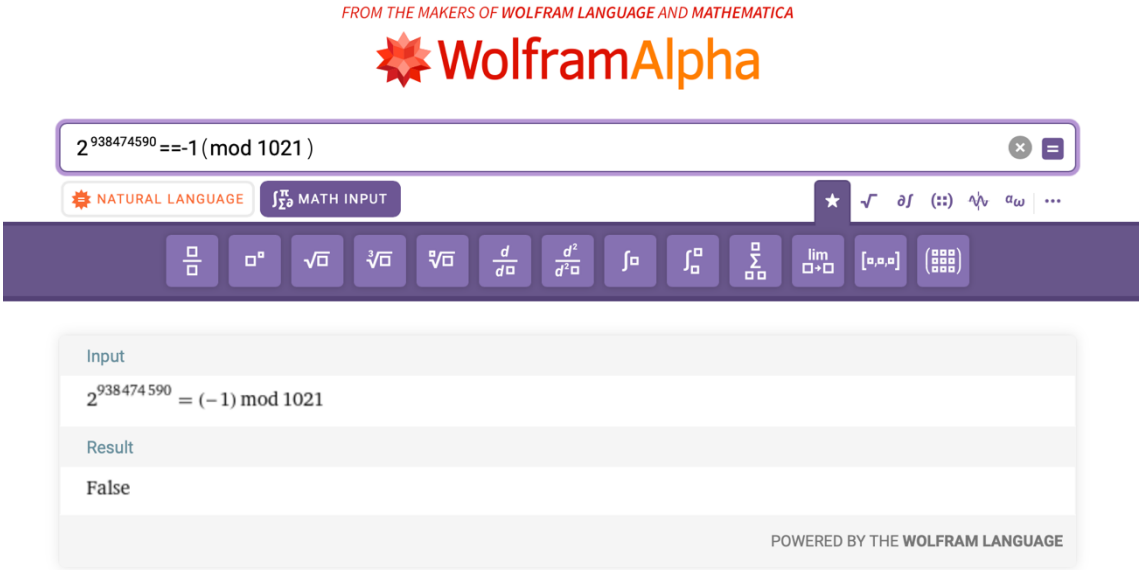
Obrázek 17: WolframAlpha Nté Mersennovo prvočíslo



Obrázek 18: WolframAlpha Ntá prvočíselná dvojčata

Wolfram Alpha je také skvělý tím, že nám ověří kongruence, které nám kalkulačka nebo Excel nezvládnou vypočítat. Stačí zadat problém přímo do vyhledávače. Kongruenci zapisujeme jako *"=="*. Abychom dostali matematické prostředí zápisu, stačí stisknout tlačítko *MATH INPUT*, které najdeme pod vyhledávacím oknem. Že jsme

v matematickém zápisu poznáme tak, že je zde fialové prostředí.

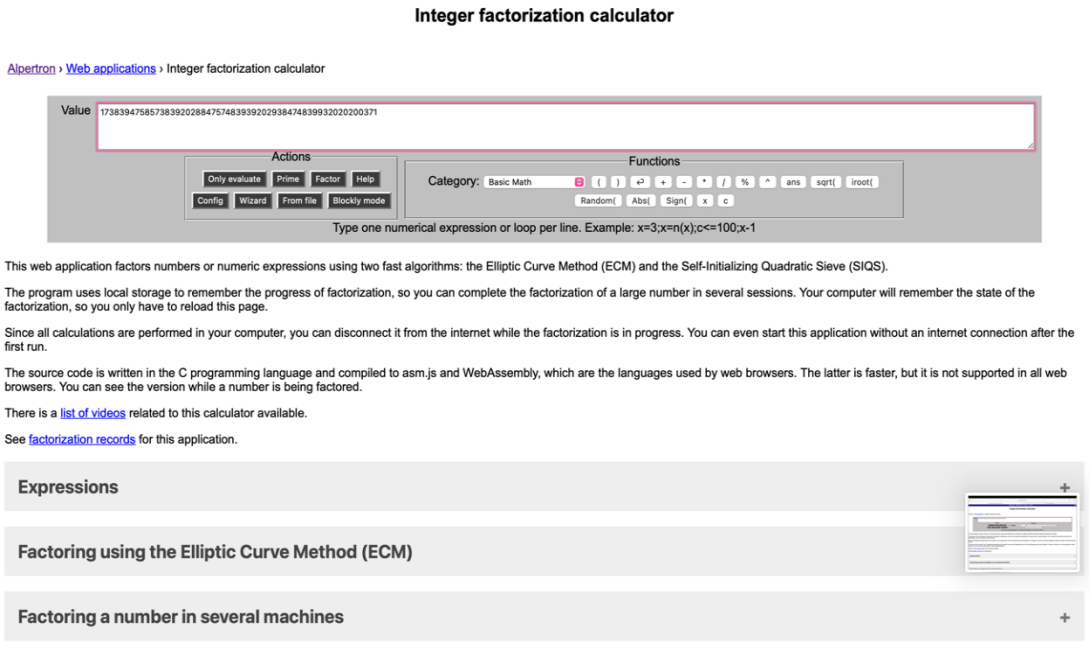


Obrázek 19: WolframAlpha ověření kongruence

9.2 Alperton

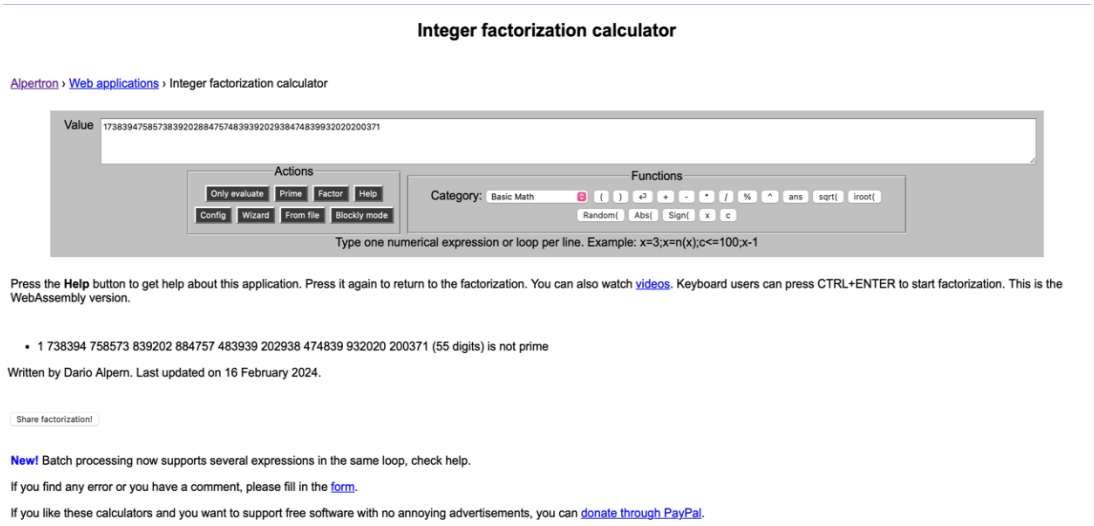
Argentinský elektroinženýr a učitel Dario Alpern vytvořil stránku, na které si můžeme ověřovat prvočísla, případně zjistit rozklad na prvočísla. Stránku nalezneme na tomto odkaze: https://www.alpertron.com.ar/ECM.HTM. Tato stránka je zcela zdarma a výsledky bývají během chvilky. Program zvládne určit prvočíselnost pro čísla s méně než 200 000 ciframi. S největším známým prvočíslem si tedy neporadí.

Do kolonky *Value* zadáme naše číslo.



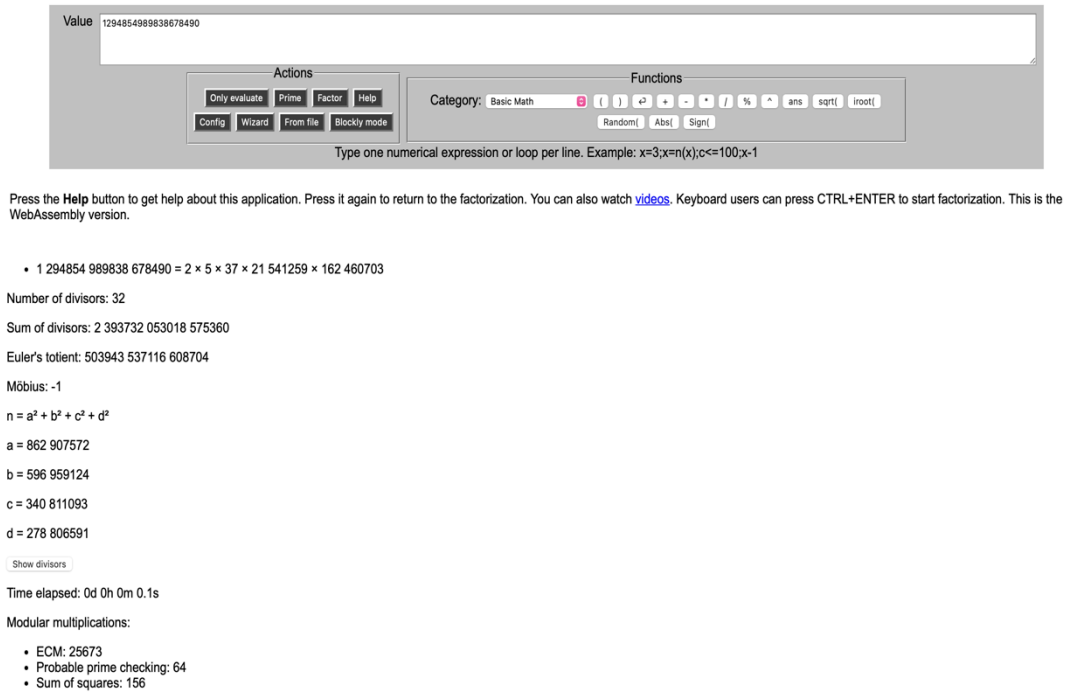
Obrázek 20: Alperton prostředí aplikace

Pokud chceme zjistit, zda se jedná o prvočísla, klikneme v záložce *Actions* na tlačítko *Prime.* Jak si můžete povšimnout, nezadali jsme příliš nízké číslo, ale i přesto se odpověď dozvíme během necelé vteřiny. Také se dozvíme, kolik číslic zadané číslo má.



Obrázek 21: Alperton prvočíselnost

Kdybychom chtěli znát rozklad na prvočísla, můžeme použít opět v *Actions* tlačítko *Factor.*



Obrázek 22: Alperton faktorizace

K faktorizaci program používá metodu eliptických křivek. Jak si můžete povšimnout, jedná se o velice rychlou metodu. Doba faktorizace devatenácticiferného čísla (obr. 22) trvala 0,1 sekundu.

9.3 Mathematica

V programu Mathematica se používá Wolfram language, což je symbolický jazyk. Pokud tento jazyk neovládáme, na stránkách https://reference.wolfram.com/language/ najdeme celou jeho dokumentaci. Všechny funkce jsou zde popsané a snadno pochopitelné.

Odlišností oproti předchozím stránkám je nutnost stažení programu a jeho zaplacení. Také prostředí už je odlišné oproti předchozím. Máme příkazy, které do programu zapisujeme, musíme u nich ovšem dodržet syntaxi, jinak si s tím program neporadí.

V prostředí se nám po levé straně budou zobrazovat malými písmeny nápisy *In[X]* a *Out[X]*. *In* je to, co do programu zadáme, tedy příkaz. A *Out* je kolonka, kde se nám zobrazí odpověď, výpočet nebo například graf. X v hranatých závorkách pouze říká pořadí příkazu, tedy kolikátý příkaz to v souboru je.

Také spuštění výpočtu je odlišné od předchozích dvou, už zde nemáme tlačítka, a tak musíme provádět výpočty jinak. To se provádí pomocí klávesové zkratky *Shift* + *Enter*.

Kontrola prvočíselnosti, faktorizace čísel

Pro kontrolu prvočíselnosti použijeme jednoduchý příkaz *PrimeQ[n]*. Pokud bude *n* prvočíslem, tak nám program vrátí hodnotu *True*, pokud ne, hodnotu *False*.

```
In[1]:= PrimeQ[1248]
Out[1]= False

In[2]:= PrimeQ[3049]
Out[2]= True
```

Obrázek 23: Mathematica – příkaz PrimeQ[n]

Pokud bychom chtěli zjistit faktorizaci, zadáme příkaz *FactorInteger[n]*. Grafické zpracování nevypadá tak dobře, jako u předchozích stránek, protože je program určen pro profesionály nebo ke vzdělávacím účelům a ne pro širokou veřejnost.

```
In[3]:= FactorInteger[2468]
Out[3]= {{2, 2}, {617, 1}}

In[4]:= FactorInteger[3049]
Out[4]= {{3049, 1}}
```

Obrázek 24: Mathematica – příkaz FactorInteger[n]

Generování prvočísel

Pro n-té prvočíslo napíšeme příkaz *Prime[n]*. Pokud bychom chtěli zjistit prvočíslo, které je po čísle n, použijeme *NextPrime[n]*.

```
In[8]:= Prime[1000]
Out[8]= 7919

In[9]:= NextPrime[100000]
Out[9]= 100003
```

Obrázek 25: Mathematica – příkazy Prime[n], NextPrime[n]

Zajímavou a velmi užitečnou funkcí je také příkaz na hledání náhodných čísel. Příkaz vypadá následovně *RandomPrime[max]*. Tento příkaz vypíše náhodné prvočíslo

z rozsahu od 2 do maximální hodnoty max. Kdyby nám nestačilo jedno jediné náhodné prvočíslo, lze zapsat příkaz ještě jedním způsobem *RandomPrime[max, n]*. Příkaz funguje podobně jako předchozí, jen nám vypíše n prvočísel. Pokud bychom chtěli rozsah zmenšit, můžeme napsat *RandomPrime[{min, max}]*. Zde program vypíše náhodné prvočíslo mezi hodnotami min a max.

```
In[5]:= RandomPrime[1000]
Out[5]= 487

In[6]:= RandomPrime[1000, 10]
Out[6]= {109, 797, 467, 67, 457, 19, 73, 449, 89, 379}

In[7]:= RandomPrime[{10000, 90000}]
Out[7]= 12541
```

Obrázek 26: Mathematica – příkazy RandomPrime

Závěr

Cílem této bakalářské práce bylo představení klasických i nových poznatků o prvočíslech. Ve většině kapitol jsou rozebrány poznatky od nejstarších a nejznámějších až po ty nejnovější a nejmodernější, které se používají například v kryptografii.

Nejpodrobněji z celé práce jsou zpracovány faktorizační algoritmy a testy, u každého tématu je ukázáno několik metod, které jsou zde vysvětleny a ukázány na příkladech, každý si tedy může vyzkoušet tu, která se mu zamlouvá. V práci jsou popsány jednodušší a používanější algoritmy, které by měly být pro většinu lidí pochopitelné.

V závěru práce jsou představeny aplikace, které se dají využít nejen pro práci s prvočísly. Takových aplikací existuje celá řada, ale byly vybrány tyto, protože je jejich prostředí uživatelsky přívětivé a byly použity pro ověřovaní a zjišťování výsledků v této práci.

Resumé

Práce je zaměřena na klasické a nové poznatky o prvočíslech. Obsahuje historii prvočísel, metody a algoritmy pro ověřování prvočíselnosti a metody pro faktorizaci přirozených čísel. Dále je zde představena prvočíselná funkce a způsoby, jakými se prvočísla hledají. Na závěr jsou představeny aplikace, které můžeme použít pro hledání a ověřování prvočísel a faktorizaci přirozených čísel.

Summary

This work is focused on classical and new knowledge about prime numbers. Contains the history of prime numbers, methods and algorithms for verifying prime numbers, and methods for factoring natural numbers. Then the prime number function and the ways in which prime numbers are searched for are introduced here. Finally, applications that we can use to find and verify prime numbers and factor natural numbers are described.

Seznam literatury

[1] Dostupní z: http://web.math.muni.cz/~fuchs/Efuchs/historie_pdf/7proc.pdf [citováno 2024-03-05]

[2] DU SAUTOY, Marcus. *Hudba prvočísel: dvě století Riemannovy hypotézy*. Přeložil Luboš PICK, přeložil Mirko ROKYTA. Zip (Argo: Dokořán). Praha: Argo, 2019. ISBN 978-80-257-2951-9. [47–97]

[3] BURDA, Karel. *Kryptografie okolo nás*. CZ.NIC. Praha: CZ.NIC, z.s. p.o., 2019. ISBN 978-80-88168-49-2. [15,21]

[4] Dostupné z: https://www.man-in-the-middle.cz [citováno 2024-03-15]

[5] STALLINGS, William. *Cryptography and network security: principles and practice*. Seventh edition. Global edition. Boston: Pearson, [2017]. ISBN 978-1-292-15858-7.

[6] Dostupné z: https://www.maths.cz/clanky/118-prvociselny-rozklad [citováno 2024-03-17]

[7] CRANDALL, Richard a POMERANCE, Carl. *Prime numbers: A Computational Perspective*. 2. vydání. Springer, 2005. ISBN 0387252827. [225-226]

[8] POMERANCE, Carl. A Tale of Two Sieves. *Notices of the American Mathematical Society*. December 1996, roč. 43, č. 12, s. 1473-1480.

[9] Dostupné z: https://kconrad.math.uconn.edu/blurbs/ugradnumthy/solovaystrassen.pdf [citováno 2024-03-20]

[10] Dostupné z: https://t5k.org/prove/prove3_1.html [citováno 2024-03-30]

[11] CRANDALL, Richard a POMERANCE, Carl. *Prime numbers: A Computational Perspective*. 2. vydání. Springer, 2005. ISBN 0387252827. [70]

[12] Dostupné z: https://kdm.karlin.mff.cuni.cz//sborniky/sbornik-30.pdf [citováno 2024-04-20]

[13] RIBENBOIM, Paulo. Selling Primes. *Mathematics Magazine*. 1995, roč. 68, č. 3, s. 175-182. Dostupné z: https://www.math.stonybrook.edu/~moira/mat331-spr10/papers/1995%20RibenboimSelling%20Primes.pdf [citováno 2024-04-04]

[14] Dostupné z: https://www.mersenne.org/various/history.php [citováno 2024-03-23]

[15] Dostupné z: https://publications.azimpremjiuniversity.edu.in/3346/1/27-Anand_PrimeGeneratingPolynomials_Final.pdf [citováno 2024-03-20]

[16] Dostupné z: https://mathworld.wolfram.com/Prime-GeneratingPolynomial.html [citováno 2024-03-20]

[17] Dostupné z: https://www.mathpuzzle.com/MAA/48-Prime%20Generating%20Polynomials/mathgames_07_17_06.html [citováno 2024-03-22]

[18] DU SAUTOY, Marcus. *Hudba prvočísel: dvě století Riemannovy hypotézy*. Přeložil Luboš PICK, přeložil Mirko ROKYTA. Zip (Argo: Dokořán). Praha: Argo, 2019. ISBN 978-80-257-2951-9. [220-221]

[19] Dostupné z: https://maa.org/sites/default/files/pdf/upload_library/22/Ford/JonesSatoWadaWiens.pdf [citováno 2024-04-13]

Seznam obrázků

Seznam tabulek