Úvod a popis

Problematikou, kterou se tento projekt zabývá je generování a ověřování velkých prvočísel. Používá k tomu 3 algoritmy (Atkinovo síto, Miller–Rabin, Postupné dělení)
Uživatel buď může zadat, jestli chce otestovat zda jde o prvočíslo nebo si může nechat vygenerovat prvočíslo po zadané číslo nebo prvočíslo, které je zadanému číslu nejbližší a následně zadá jaký algoritmus k tomu chce použít.
Prvočísla jsou buď vypisována na obrazovku nebo do souboru (každý řádek je jedno prvočíslo)
Je také možné načítat prvočísla ze souboru, kde zase jako u ukládání je jedno prvočíslo na jeden řádek.
Události jako začátek generování, stav generovaní, chyby, atd… jsou ukládány do logu.
Čas, který byl potřeba pro vygenerování nebo otestování prvočísla je ukládán do samostatného souboru ve, kterém jsou uloženy údaje: začáteční čas a datum generovaní;čas na generovaní;zkrácené prvočíslo (když je prvočíslo moc dlouhé je zapsáno ve vědeckém tvaru)

Cíle projektu

- Generovat a testovat prvočísla s teoreticky neomezenou velikostí
- Načítat a ukládat prvočísla ze souborů nebo z a do terminálu
- Ukládat do souboru prvocisla_cas.log jak dlouho trvalo generovaní nebo ověřování prvočísel
- Ukládat do souboru udalosti.log právě probíhající události

Teoretická část

Prvočíslo

Definice

Prvočíslo je přirozené číslo větší než 1, které je beze zbytku dělitelné jen dvěma děliteli: jedničkou a samo sebou. Jednička není prvočíslo, neboť nemá dva různé dělitele. Přirozená čísla větší než jedna, která nejsou prvočísly, se nazývají složená čísla. Prvním prvočíslem je číslo 2, které je jediným sudým prvočíslem.

Prvočísla do 1000

2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97, 101, 103, 107, 109, 113, 127, 131, 137, 139, 149, 151, 157, 163, 167, 173, 179, 181, 191, 193, 197, 199, 211, 223, 227, 229, 233, 239, 241, 251, 257, 263, 269, 271, 277, 281, 283, 293, 307, 311, 313, 317, 331, 337, 347, 349, 353, 359, 367, 373, 379, 383, 389, 397, 401, 409, 419, 421, 431, 433, 439, 443, 449, 457, 461, 463, 467, 479, 487, 491, 499, 503, 509, 521, 523, 541, 547, 557, 563, 569, 571, 577, 587, 593, 599, 601, 607, 613, 617, 619, 631, 641, 643, 647, 653, 659, 661, 673, 677, 683, 691, 701, 709, 719, 727, 733, 739, 743, 751, 757, 761, 769, 773, 787, 797, 809, 811, 821, 823, 827, 829, 839, 853, 857, 859, 863, 877, 881, 883, 887, 907, 911, 919, 929, 937, 941, 947, 953, 967, 971, 977, 983, 991, 997

Největší prvočíslo

Největší známé prvočíslo je $2^{136279841} - 1$.
Číslo bylo objeveno v rámci projektu GIMPS v říjnu 2024.

Algoritmy

Atkinovo síto

Jedná se o moderní algoritmus, který dokáže najít prvočísla až po zadané celé číslo. V porovnání se starověkým Eratosthenovým sítem, které odděluje násobky prvočísel, Atkinovo síto provádí předběžnou práci a poté odděluje násobky odmocnin prvočísel, čímž dosahuje lepší teoretické asymptotické složitosti. V roce 2003 jej vytvořili A. O. L. Atkin a Daniel J. Bernstein.

Algoritmus

V algoritmu:

- Všechny zbytky jsou modulo 60
- Všechny čísla i, x a y jsou kladné celé čísla

Postup:

1. Vytvořte list výsledků naplněný s 2,3,5
2. Vytvořte seznam síta s položkou pro každé kladné celé číslo; všechny položky tohoto seznamu by měly být zpočátku označeny jako složené.
3. Pro každou položku číslo n v seznamu síta s modulo se zbytkem r modulo 60:
    1. Je-li r 1, 13, 17, 29, 37, 41, 49 nebo 53, převraťte zadání pro každé možné řešení $4x^2 + y^2 = n$. Počet operací převracení v poměru k rozsahu síta pro tento krok se blíží $\frac{4\sqrt{\pi}}{15} \cdot \frac{8}{60}$ („8“ ve zlomku pochází z osmi modulů, které tato kvadratura zpracovává, a 60 proto, že Atkin to vypočítal na základě sudého počtu modulo 60 kol), což vede ke zlomku přibližně 0,1117010721276..…
    2. Pokud je r 7, 19, 31 nebo 43, převraťte zadání pro každé možné řešení $3x^2 + y^2 = n$. Počet operací převracení v poměru k rozsahu síta pro tento krok se blíží $\pi\sqrt{0,12} \times 4/60$ („4“ ve zlomku pochází ze čtyř modulů, které tato kvadratura zpracovává, a 60 proto, že Atkin to vypočítal na základě sudého počtu modulo 60 kol), což vede ke zlomku přibližně 0,072551974569..…
    3. Je-li r 11, 23, 47 nebo 59, převraťte vstup pro každé možné řešení $3x^2 - y^2 = n$, když x > y. Počet operací převracení v poměru k rozsahu sítování pro tento krok se blíží $\sqrt{1,92} \ln(\sqrt{0,5} + \sqrt{1.5}) \times 4/60$ (číslo „4“ ve zlomku pochází ze čtyř modulů, které tato kvadratura zpracovává, a číslo 60 proto, že Atkin to vypočítal na základě sudého počtu modulo 60 kol), což vede ke zlomku přibližně 0,060827679704.…
    4. Pokud je r něco jiného tak ho ignorujeme
4. Začneme s nejnižším číslem v seznamu síta.
5. Vezměte další číslo v seznamu síta, které je ještě označeno jako prvočíslo.
6. Toto číslo zahrňte do seznamu výsledků.
7. Toto číslo odmocněte a všechny jeho násobky označte jako neprimární.
   Všimněte si, že násobky, které lze vynásobit číslem 2, 3 nebo 5, není třeba označovat, protože ty budou v konečném výčtu prvočísel ignorovány.

8. Opakujte kroky 4 až 7. Celkový počet operací při těchto opakováních označování čtverců prvočísel jako poměru rozsahu sítování je součet inverzních hodnot čtverců prvočísel, které se blíží funkci prvočísla zeta na 2 (rovná se 0,45224752004...) mínus $1/2^2$, $1/3^2$ a $1/5^2$ pro ta prvočísla, která byla vyřazena kolem, přičemž výsledek se vynásobí 16/60 pro poměr zásahů kola na rozsah; výsledkem je poměr přibližně 0,01363637571..…

Vysvětlení

Algoritmus zcela ignoruje jakákoli čísla se zbytkem modulo 60, která jsou dělitelná 2, 3 nebo 5, protože čísla se zbytkem modulo 60 dělitelným jedním z těchto tří prvočísel jsou sama dělitelná tímto prvočíslem.

Všechna čísla n se zbytkem po modulu šedesát 1, 13, 17, 29, 37, 41, 49 nebo 53 mají zbytek po modulu čtyři 1. Tato čísla jsou prvočísla tehdy a jen tehdy, když počet řešení úlohy $4x^2 + y^2 = n$ je lichý a číslo je bez odmocniny.

Všechna čísla n s modulo-šedesátým zbytkem 7, 19, 31 nebo 43 mají modulo-šestý zbytek 1. Tato čísla jsou prvočísla tehdy a jen tehdy, je-li počet řešení úlohy $3x^2 + y^2 = n$ lichý a číslo je bez odmocniny.

Všechna čísla n s modulo 60 zbytkem 11, 23, 47 nebo 59 mají modulo-dvanáctkový zbytek 11. Tato čísla jsou prvočísla tehdy a jen tehdy, je-li počet řešení $3x^2 - y^2 = n$ lichý a číslo je bez odmocnin.

Teoretická časová náročnost

Časová náročnost byla měla být O(N log(log(N)))

Miller–Rabin

Test Miller–Rabin je algoritmus používaný k ověření, zda je dané číslo prvočíslem. Jedná se o rozšíření Fermatova testu prvočíselnosti, které poskytuje spolehlivější výsledky. Původní verzi tohoto testu navrhl Gary L. Miller jako deterministický algoritmus za předpokladu platnosti rozšířené Riemannovy hypotézy. Později Michael O. Rabin tuto metodu upravil na pravděpodobnostní algoritmus, který nevyžaduje neprokázané předpoklady.

Princip testu:

Test Miller–Rabin využívá skutečnosti, že pro každé prvočíslo n platí určitá vlastnost týkající se mocnin náhodně zvoleného čísla a (nazývaného báze). Konkrétně, pokud je n prvočíslo, pak pro každé a platí:

1. $a^d \equiv 1 \pmod n$ nebo
2. Existuje celé číslo r ($0 \le r < s$), pro které platí $a^{2^r \cdot d} \equiv -1 \pmod n$

kde $n-1=2^s \cdot d$ a **d** je liché číslo. Pokud ani jedna z těchto podmínek není splněna, číslo n je složené a **a** je svědek složenosti **n**.

Teoretická časová složitost

Doba běhu tohoto algoritmu je O(log n), pro n-ciferné číslo, přičemž k je počet provedených kol; jedná se tedy o efektivní algoritmus s polynomiálním časem.

Pravděpodobnostní charakter testu:

Test Miller–Rabin je pravděpodobnostní, což znamená, že při opakovaném testování s různými bázemi a lze snížit pravděpodobnost falešně pozitivního výsledku (tj. kdy složené číslo je označeno jako prvočíslo). Konkrétně, pokud je n složené, pravděpodobnost, že náhodně zvolená báze a nebude svědkem složenosti, je menší než 1/4. Tím pádem, po k nezávislých opakováních testu je pravděpodobnost, že n je složené, ale test to neodhalil, menší než (1/4)k.

Deterministické varianty:

Existují deterministické varianty testu Miller–Rabin, které zaručují správný výsledek bez pravděpodobnostního charakteru. Tyto varianty testují číslo n pomocí předem stanovené množiny bází. Například:

- když n < 2,047- stačí testovat a = 2;
- když n < 1,373,653- stačí testovat a = 2, 3;
- když n < 9,080,191- stačí testovat a = 31, 73;
- když n < 25,326,001- stačí testovat a = 2, 3, 5;
- když n < 3,215,031,751- stačí testovat a = 2, 3, 5, 7;
- když n < 4,759,123,141- stačí testovat a = 2, 7, 61;
- když n < 1,122,004,669,633- stačí testovat a = 2, 13, 23, 1662803;
- když n < 2,152,302,898,747- stačí testovat a = 2, 3, 5, 7, 11;
- když n < 3,474,749,660,383- stačí testovat a = 2, 3, 5, 7, 11, 13;
- když n < 341,550,071,728,321- stačí testovat a = 2, 3, 5, 7, 11, 13, 17.

Tyto předem stanovené báze umožňují efektivní a spolehlivé testování prvočíselnosti pro čísla v daném rozsahu.

Postupné dělení

Postupné dělení je ze všech algoritmů faktorizace celých čísel nejpracnější, ale nejsnadněji pochopitelný. Základní myšlenka postupného dělení spočívá v tom, že se testuje, zda lze celé číslo n vydělit číslem v pořadí, které je menší nebo rovno druhé odmocnině z n.

Příklad - kde není prvočíslo

$n = 49 \sqrt{n} = 7$

- Je 7 ≡ 0 mod 2 → ne
- Je 7 ≡ 0 mod 3 → ne
- Je 7 ≡ 0 mod 4 → ne
- ….
- Je 7 ≡ 0 mod 7 → **ano**

Odmocnina n je dělitelná 7 tím pádem číslo není prvočíslem

Příklad - kde je prvočíslo

$n = 103 \sqrt{n} = 10{,}148891565$

- Je 10,148891565 ≡ 0 mod 2 → ne
- Je 10,148891565 ≡ 0 mod 3 → ne
- Je 10,148891565 ≡ 0 mod 4 → ne
- ….
- Je 10,148891565 ≡ 0 mod 9 → ne

- Je 10,148891565 ≡ 0 mod 10 → ne

Odmocnina n není dělitelná žádným číslem z čehož vychází, že jde o prvočíslo.

Teoretická časová náročnost

O(√n/log n)

stav projektu

Projekt se nachází v převážně v teoretické části.

Ještě je třeba naprogramovat funkční program s danými algoritmy a otestovat jak rychle a jak velké číslo program dokáže vygenerovat a otestovat.

Autoři projektu a přínos do projektu

Marek Roháč – Napsání studie, Příprava teorie

Matyáš Vilím – Programování kódu, Testování a měření algoritmů

Kryštof Troják – Programování kódu, Příprava teorie

Literatura

Atkin, A.O.L.; Bernstein, D.J. Prime sieves using binary quadratic forms. Mathematics of Computation. 2004, roč. 73, č. 246, s. 1023–1030. Dostupné z: https://www.ams.org/journals/mcom/2004-73-246/S0025-5718-03-01501-1/S0025-5718-03-01501-1.pdf

Trial division. Wikipedia: The Free Encyclopedia [online]. Dostupné z: https://en.wikipedia.org/wiki/Trial_division

Matematické základy kryptografických algoritmů. Vysoká škola báňská – Technická univerzita Ostrava [online]. Dostupné z: https://mi21.vsb.cz/sites/mi21.vsb.cz/files/unit/matem_zaklady_kryptograf_algoritmu_obr.pdf

GIMPS. 52nd Known Mersenne Prime Discovered [online]. 2024. Dostupné z: https://www.mersenne.org/primes/?press=M136279841

Johnstone, I.M.; Lundy, P.J. A new Mersenne prime. Journal of the Australian Mathematical Society. 1980, roč. 29, č. 2, s. 149–153. Dostupné z: https://www.sciencedirect.com/science/article/pii/0022314X80900840?via%3Dihub

The Prime Pages. Prime number research, records and results [online]. Dostupné z: https://t5k.org/

Schoof, R. Four primality testing algorithms. 2004. Dostupné z: https://www.mat.uniroma2.it/~schoof/millerrabinpom.pdf

Miller–Rabin primality test. Wikipedia: The Free Encyclopedia [online]. Dostupné z: https://en.wikipedia.org/wiki/Miller%E2%80%93Rabin_primality_test