NAJNUJNEJŠE O TESTIH PRAŠTEVILSKOSTI

Aleksander Simonič

Fakulteta za matematiko, naravoslovje in informacijske tehnologije
Univerza na Primorskem

18/03/2026

Praštevila in njihove lastnosti

Definicija 1

Naravno število $n \geq 2$ *se imenuje* ***praštevilo****, če ima* n *natanko dva pozitivna delitelja, to sta* 1 *in* n*. Ostala naravna števila* ≥ 2 *se imenujejo* ***sestavljena****.*

Prvih deset praštevil:

$$2, 3, 5, 7, 11, 13, 17, 19, 23, 29.$$

Nekaj osnovnih vprašanj:

1. Zakaj so praštevila pomembna?
2. Koliko je vseh praštevil?
3. Kako določimo, ali je dano naravno število praštevilo?

Citat iz *Disquisitiones Arithmeticae*:

The problem of distinguishing prime numbers from composite numbers and of resolving the latter into their prime factors is known to be one of the most important and useful in arithmetic. It has engaged the industry and wisdom of ancient and modern geometers to such an extent that it would be superfluous to discuss the problem at length. Nevertheless we must confess that all methods that have been proposed thus far are either restricted to very special cases or are so laborious and prolix that even for numbers that do not exceed the limits of tables constructed by estimable men, i.e., for numbers that do not yield to artificial methods, they try the patience of even the practiced calculator. And these methods do not apply at all to larger numbers.

Praštevila in njihove lastnosti

Definicija 1

*Naravno število $n \geq 2$ se imenuje **praštevilo**, če ima n natanko dva pozitivna delitelja, to sta 1 in n. Ostala naravna števila ≥ 2 se imenujejo **sestavljena**.*

Prvih deset praštevil:

$$2, 3, 5, 7, 11, 13, 17, 19, 23, 29.$$

Nekaj osnovnih vprašanj:

1. Zakaj so praštevila pomembna?
2. Koliko je vseh praštevil?
3. Kako določimo, ali je dano naravno število praštevilo?

Citat iz *Disquisitiones Arithmeticae*:

The problem of distinguishing prime numbers from composite numbers and of resolving the latter into their prime factors is known to be one of the most important and useful in arithmetic. It has engaged the industry and wisdom of ancient and modern geometers to such an extent that it would be superfluous to discuss the problem at length. Nevertheless we must confess that all methods that have been proposed thus far are either restricted to very special cases or are so laborious and prolix that even for numbers that do not exceed the limits of tables constructed by estimable men, i.e., for numbers that do not yield to artificial methods, they try the patience of even the practiced calculator. And these methods do not apply at all to larger numbers.

Praštevila in njihove lastnosti

Definicija 1

*Naravno število $n \geq 2$ se imenuje **praštevilo**, če ima n natanko dva pozitivna delitelja, to sta 1 in n. Ostala naravna števila ≥ 2 se imenujejo **sestavljena**.*

Prvih deset praštevil:

$$2, 3, 5, 7, 11, 13, 17, 19, 23, 29.$$

Nekaj osnovnih vprašanj:

1. Zakaj so praštevila pomembna?
2. Koliko je vseh praštevil?
3. Kako določimo, ali je dano naravno število praštevilo?

Citat iz *Disquisitiones Arithmeticae*:

The problem of distinguishing prime numbers from composite numbers and of resolving the latter into their prime factors is known to be one of the most important and useful in arithmetic. It has engaged the industry and wisdom of ancient and modern geometers to such an extent that it would be superfluous to discuss the problem at length. Nevertheless we must confess that all methods that have been proposed thus far are either restricted to very special cases or are so laborious and prolix that even for numbers that do not exceed the limits of tables constructed by estimable men, i.e., for numbers that do not yield to artificial methods, they try the patience of even the practiced calculator. And these methods do not apply at all to larger numbers.

Praštevila in njihove lastnosti

Praštevila imamo lahko za osnovne gradnike vseh ostalih naravnih števil ≥ 2.

Evklid (okoli 300 pr. n. š.).

Izrek 1

Naj bo $n \geq 2$ naravno število. Potem obstaja tako praštevilo p, da je število n deljivo s p.

Izrek 2

Obstaja neskončno mnogo praštevil.

Carl F. Gauss (1777–1855).

Izrek 3 (Osnovni izrek aritmetike)

Naj bo $n \geq 2$ naravno število. Potem obstajajo praštevila $p_1 < \cdots < p_k$ in naravna števila $m_1, \ldots, m_k$, da velja $n = p_1^{m_1} \cdots p_k^{m_k}$. Ta zapis je pri pogoju $p_1 < \cdots < p_k$ enoličen.

Osnovni izrek aritmetike zagotavlja *prafaktorizacijo* poljubnega naravnega števila ≥ 2, npr.

$$1836765 = 3^2 \cdot 5 \cdot 7^4 \cdot 17.$$

Praštevila in njihove lastnosti

Praštevila imamo lahko za osnovne gradnike vseh ostalih naravnih števil ≥ 2.

Evklid (okoli 300 pr. n. š.).

Izrek 1

Naj bo $n \geq 2$ naravno število. Potem obstaja tako praštevilo p, da je število n deljivo s p.

Izrek 2

Obstaja neskončno mnogo praštevil.

Carl F. Gauss (1777–1855).

Izrek 3 (Osnovni izrek aritmetike)

Naj bo $n \geq 2$ naravno število. Potem obstajajo praštevila $p_1 < \cdots < p_k$ in naravna števila $m_1, \ldots, m_k$, da velja $n = p_1^{m_1} \cdots p_k^{m_k}$. Ta zapis je pri pogoju $p_1 < \cdots < p_k$ enoličen.

Osnovni izrek aritmetike zagotavlja *prafaktorizacijo* poljubnega naravnega števila ≥ 2, npr.

$$1836765 = 3^2 \cdot 5 \cdot 7^4 \cdot 17.$$

Praštevila in njihove lastnosti

Praštevila imamo lahko za osnovne gradnike vseh ostalih naravnih števil ≥ 2.

Evklid (okoli 300 pr. n. š.).

Izrek 1

Naj bo $n \geq 2$ naravno število. Potem obstaja tako praštevilo p, da je število n deljivo s p.

Izrek 2

Obstaja neskončno mnogo praštevil.

Carl F. Gauss (1777–1855).

Izrek 3 (Osnovni izrek aritmetike)

Naj bo $n \geq 2$ naravno število. Potem obstajajo praštevila $p_1 < \cdots < p_k$ in naravna števila $m_1, \ldots, m_k$, da velja $n = p_1^{m_1} \cdots p_k^{m_k}$. Ta zapis je pri pogoju $p_1 < \cdots < p_k$ enoličen.

Osnovni izrek aritmetike zagotavlja *prafaktorizacijo* poljubnega naravnega števila ≥ 2, npr.

$$1836765 = 3^2 \cdot 5 \cdot 7^4 \cdot 17.$$

Praštevila in njihove lastnosti

Praštevila imamo lahko za osnovne gradnike vseh ostalih naravnih števil ≥ 2.

Evklid (okoli 300 pr. n. š.).

Izrek 1

Naj bo $n \geq 2$ naravno število. Potem obstaja tako praštevilo p, da je število n deljivo s p.

Izrek 2

Obstaja neskončno mnogo praštevil.

Carl F. Gauss (1777–1855).

Izrek 3 (Osnovni izrek aritmetike)

Naj bo $n \geq 2$ naravno število. Potem obstajajo praštevila $p_1 < \cdots < p_k$ in naravna števila $m_1, \ldots, m_k$, da velja $n = p_1^{m_1} \cdots p_k^{m_k}$. Ta zapis je pri pogoju $p_1 < \cdots < p_k$ enoličen.

Osnovni izrek aritmetike zagotavlja *prafaktorizacijo* poljubnega naravnega števila ≥ 2, npr.

$$1836765 = 3^2 \cdot 5 \cdot 7^4 \cdot 17.$$

Praštevila in njihove lastnosti

Evklid je dokazal, da je praštevil neskončno. Kljub temu se lahko vprašamo po njihovi *gostoti*. Naj bo $\pi(x)$ število praštevil, ki ne presegajo realnega števila $x \geq 2$, npr. $\pi(2) = 2$ in $\pi(10.8) = 4$. Očitno je $\pi(x) \leq x$.

Gauss in **Adrien-Marie Legendre** (1752–1833) okoli 1800.

Problem 1

Ali velja

$$\lim_{x\to\infty} \frac{\pi(x)\log x}{x} = 1?$$

Z drugimi besedami, ali za vsak $\varepsilon > 0$ obstaja $x_0 \geq 2$, da je

$$\frac{(1-\varepsilon)x}{\log x} \leq \pi(x) \leq \frac{(1+\varepsilon)x}{\log x}$$

za vse $x \geq x_0$?

Lengendre je dokazal, da velja $\lim_{x\to\infty} \pi(x)/x = 0$.

PRAŠTEVILA IN NJIHOVE LASTNOSTI

Evklid je dokazal, da je praštevil neskončno. Kljub temu se lahko vprašamo po njihovi *gostoti*. Naj bo $\pi(x)$ število praštevil, ki ne presegajo realnega števila $x \geq 2$, npr. $\pi(2) = 2$ in $\pi(10.8) = 4$. Očitno je $\pi(x) \leq x$.

Gauss in **Adrien-Marie Legendre** (1752–1833) okoli 1800.

Problem 1

Ali velja

$$\lim_{x\to\infty} \frac{\pi(x)\log x}{x} = 1?$$

Z drugimi besedami, ali za vsak $\varepsilon > 0$ obstaja $x_0 \geq 2$, da je

$$\frac{(1-\varepsilon)x}{\log x} \leq \pi(x) \leq \frac{(1+\varepsilon)x}{\log x}$$

za vse $x \geq x_0$?

Lengendre je dokazal, da velja $\lim_{x\to\infty} \pi(x)/x = 0$.

PRAŠTEVILA IN NJIHOVE LASTNOSTI

Evklid je dokazal, da je praštevil neskončno. Kljub temu se lahko vprašamo po njihovi *gostoti*. Naj bo $\pi(x)$ število praštevil, ki ne presegajo realnega števila $x \geq 2$, npr. $\pi(2) = 2$ in $\pi(10.8) = 4$. Očitno je $\pi(x) \leq x$.

Gauss in **Adrien-Marie Legendre** (1752–1833) okoli 1800.

Problem 1

Ali velja

$$\lim_{x\to\infty} \frac{\pi(x)\log x}{x} = 1?$$

Z drugimi besedami, ali za vsak $\varepsilon > 0$ obstaja $x_0 \geq 2$, da je

$$\frac{(1-\varepsilon)x}{\log x} \leq \pi(x) \leq \frac{(1+\varepsilon)x}{\log x}$$

za vse $x \geq x_0$?

Lengendre je dokazal, da velja $\lim_{x\to\infty} \pi(x)/x = 0$.

Praštevila in njihove lastnosti

Gaussova in Legendrova domneva je bila znana kot *praštevilski zakon*. Dokazana je bila šele leta 1896.

Jacques S. Hadamard (1865–1963) in **Charles-Jean de la Vallée Poussin** (1866–1962).

Izrek 4 (Praštevilski izrek)

Velja

$$\lim_{x\to\infty} \frac{\pi(x)\log x}{x} = 1.$$

Njun dokaz uporablja orodja iz *kompleksne analize* za študij *Riemannove funkcije zeta*

$$\zeta(s) = \sum_{n=1}^{\infty} \frac{1}{n^s} = \prod_p \frac{1}{1-p^{-s}}, \quad \Re\{s\} > 1.$$

G. F. Bernhard Riemann (1826–1866) leta 1859.

Izrek 5

Funkcijo $\zeta(s)$ je moč analitično razširiti na $\mathbb{C}\setminus\{1\}$. V točki $s = 1$ ima razširjena funkcija enostaven pol z ostankom 1.

Praštevila in njihove lastnosti

Gaussova in Legendrova domneva je bila znana kot *praštevilski zakon*. Dokazana je bila šele leta 1896.

Jacques S. Hadamard (1865–1963) in **Charles-Jean de la Vallée Poussin** (1866–1962).

Izrek 4 (Praštevilski izrek)

Velja

$$\lim_{x\to\infty} \frac{\pi(x)\log x}{x} = 1.$$

Njun dokaz uporablja orodja iz *kompleksne analize* za študij *Riemannove funkcije zeta*

$$\zeta(s) = \sum_{n=1}^{\infty} \frac{1}{n^s} = \prod_p \frac{1}{1-p^{-s}}, \quad \Re\{s\} > 1.$$

G. F. Bernhard Riemann (1826–1866) leta 1859.

Izrek 5

Funkcijo $\zeta(s)$ je moč analitično razširiti na $\mathbb{C} \setminus \{1\}$. V točki $s = 1$ ima razširjena funkcija enostaven pol z ostankom 1.

Praštevila in njihove lastnosti

Gaussova in Legendrova domneva je bila znana kot *praštevilski zakon*. Dokazana je bila šele leta 1896.

Jacques S. Hadamard (1865–1963) in **Charles-Jean de la Vallée Poussin** (1866–1962).

Izrek 4 (Praštevilski izrek)

Velja

$$\lim_{x\to\infty} \frac{\pi(x)\log x}{x} = 1.$$

Njun dokaz uporablja orodja iz *kompleksne analize* za študij *Riemannove funkcije zeta*

$$\zeta(s) = \sum_{n=1}^{\infty} \frac{1}{n^s} = \prod_p \frac{1}{1-p^{-s}}, \quad \Re\{s\} > 1.$$

G. F. Bernhard Riemann (1826–1866) leta 1859.

Izrek 5

Funkcijo $\zeta(s)$ je moč analitično razširiti na $\mathbb{C}\setminus\{1\}$. V točki $s = 1$ ima razširjena funkcija enostaven pol z ostankom 1.

Praštevila in njihove lastnosti

Gaussova in Legendrova domneva je bila znana kot *praštevilski zakon*. Dokazana je bila šele leta 1896.

Jacques S. Hadamard (1865–1963) in **Charles-Jean de la Vallée Poussin** (1866–1962).

Izrek 4 (Praštevilski izrek)

Velja

$$\lim_{x\to\infty} \frac{\pi(x)\log x}{x} = 1.$$

Njun dokaz uporablja orodja iz *kompleksne analize* za študij *Riemannove funkcije zeta*

$$\zeta(s) = \sum_{n=1}^{\infty} \frac{1}{n^s} = \prod_p \frac{1}{1-p^{-s}}, \quad \Re\{s\} > 1.$$

G. F. Bernhard Riemann (1826–1866) leta 1859.

Izrek 5

Funkcijo $\zeta(s)$ je moč analitično razširiti na $\mathbb{C} \setminus \{1\}$. V točki $s = 1$ ima razširjena funkcija enostaven pol z ostankom 1.

Hadamard in de la Vallée Poussin sta dokazala *praštevilski izrek z oceno napake*:

$$\pi(x) = \int_2^x \frac{\mathrm{d}u}{\log u} + O\left(x \exp\left(-C\sqrt{\log x}\right)\right)$$

za neko konstanto $C > 0$.

Ivan M. Vinogradov (1891–1983) in **Nikolai M. Korobov** (1917–2004):

$$\pi(x) = \int_2^x \frac{\mathrm{d}u}{\log u} + O\left(x \exp\left(-C\,(\log x)^{3/5}\,(\log\log x)^{-1/5}\right)\right)$$

za neko konstanto $C > 0$.

N. F. Helge von Koch (1870–1924) leta 1901.
Pravilnost *Riemannove domneve* zagotavlja

$$\pi(x) = \int_2^x \frac{\mathrm{d}u}{\log u} + O\left(\sqrt{x}\log x\right).$$

Hadamard in de la Vallée Poussin sta dokazala *praštevilski izrek z oceno napake*:

$$\pi(x) = \int_2^x \frac{\mathrm{d}u}{\log u} + O\left(x \exp\left(-C\sqrt{\log x}\right)\right)$$

za neko konstanto $C > 0$.

Ivan M. Vinogradov (1891–1983) in **Nikolai M. Korobov** (1917–2004):

$$\pi(x) = \int_2^x \frac{\mathrm{d}u}{\log u} + O\left(x \exp\left(-C\,(\log x)^{3/5}\,(\log\log x)^{-1/5}\right)\right)$$

za neko konstanto $C > 0$.

N. F. Helge von Koch (1870–1924) leta 1901.
Pravilnost *Riemannove domneve* zagotavlja

$$\pi(x) = \int_2^x \frac{\mathrm{d}u}{\log u} + O\left(\sqrt{x}\log x\right).$$

Praštevila in njihove lastnosti

Hadamard in de la Vallée Poussin sta dokazala *praštevilski izrek z oceno napake*:

$$\pi(x) = \int_2^x \frac{\mathrm{d}u}{\log u} + O\left(x \exp\left(-C\sqrt{\log x}\right)\right)$$

za neko konstanto $C > 0$.

Ivan M. Vinogradov (1891–1983) in **Nikolai M. Korobov** (1917–2004):

$$\pi(x) = \int_2^x \frac{\mathrm{d}u}{\log u} + O\left(x \exp\left(-C\left(\log x\right)^{3/5}\left(\log\log x\right)^{-1/5}\right)\right)$$

za neko konstanto $C > 0$.

N. F. Helge von Koch (1870–1924) leta 1901.
Pravilnost *Riemannove domneve* zagotavlja

$$\pi(x) = \int_2^x \frac{\mathrm{d}u}{\log u} + O\left(\sqrt{x}\log x\right).$$

RSA KRIPTOSISTEM

To je kriptosistem z javnim ključem, razvit leta 1977: **Ronald L. Rivest**, **Adi Shamir** in **Leonard Adleman**.



Algorithm 6.4: RSA PARAMETER GENERATION

1. Generate two large primes, p and q, such that $p \neq q$
2. $n \leftarrow pq$ and $\phi(n) \leftarrow (p-1)(q-1)$
3. Choose a random b $(1 < b < \phi(n))$ such that $\gcd(b, \phi(n)) = 1$
4. $a \leftarrow b^{-1} \bmod \phi(n)$
5. The public key is (n, b) and the private key is (p, q, a).

Povzeto po D. R. Stinson, M. B. Paterson, *Cryptography: theory and practice*, 4th. ed., Textbooks in Mathematics, CRC Press, 2019.

TEST PRAŠTEVILSKOSTI

Želimo najti *učinkovite* algoritme (to so taki algoritmi, pri katerih je časovna zahtevnost polinomska funkcija na parametru *velikosti* vhodnega podatka), ki bi nam rešili naslednja odločitvena problema:

1. PRAŠTEVILA: Za podano naravno število $n \geq 2$, ali je n praštevilo?
2. SESTAVLJENA ŠTEVILA: Za podano naravno število $n \geq 2$, ali je n sestavljeno število?

Poglejmo si osnovnošolsko metodo. Naj bo $n \geq 4$ dano naravno število. Očiten postopek za prejšnja problema bi bil, da bi preverili, ali katero od števil $d \in \{2, \ldots, \lfloor n/2 \rfloor\}$ deli n. Če tak delitelj obstaja, potem je n sestavljeno število, drugače pa je n praštevilo.

Program, napisan v jeziku *Python*:

```
def simple_test(n):
    # n>2 is odd integer
    d = 2
    while d <= n//2:
        if n%d == 0:
            return f"{n} is composite"
        else:
            d = d + 1
    return f"{n} is prime"
```

TEST PRAŠTEVILSKOSTI

Želimo najti *učinkovite* algoritme (to so taki algoritmi, pri katerih je časovna zahtevnost polinomska funkcija na parametru *velikosti* vhodnega podatka), ki bi nam rešili naslednja odločitvena problema:

1. PRAŠTEVILA: Za podano naravno število $n \geq 2$, ali je n praštevilo?
2. SESTAVLJENA ŠTEVILA: Za podano naravno število $n \geq 2$, ali je n sestavljeno število?

Poglejmo si osnovnošolsko metodo. Naj bo $n \geq 4$ dano naravno število. Očiten postopek za prejšnja problema bi bil, da bi preverili, ali katero od števil $d \in \{2, \ldots, \lfloor n/2 \rfloor\}$ deli n. Če tak delitelj obstaja, potem je n sestavljeno število, drugače pa je n praštevilo.

Program, napisan v jeziku *Python*:

```
def simple_test(n):
    # n>2 is odd integer
    d = 2
    while d <= n//2:
        if n%d == 0:
            return f"{n} is composite"
        else:
            d = d + 1
    return f"{n} is prime"
```

TEST PRAŠTEVILSKOSTI

Želimo najti *učinkovite* algoritme (to so taki algoritmi, pri katerih je časovna zahtevnost polinomska funkcija na parametru *velikosti* vhodnega podatka), ki bi nam rešili naslednja odločitvena problema:

1. PRAŠTEVILA: Za podano naravno število $n \geq 2$, ali je n praštevilo?
2. SESTAVLJENA ŠTEVILA: Za podano naravno število $n \geq 2$, ali je n sestavljeno število?

Poglejmo si osnovnošolsko metodo. Naj bo $n \geq 4$ dano naravno število. Očiten postopek za prejšnja problema bi bil, da bi preverili, ali katero od števil $d \in \{2, \ldots, \lfloor n/2 \rfloor\}$ deli n. Če tak delitelj obstaja, potem je n sestavljeno število, drugače pa je n praštevilo.

Program, napisan v jeziku *Python*:

```
def simple_test(n):
    # n>2 is odd integer
    d = 2
    while d <= n//2:
        if n%d == 0:
            return f"{n} is composite"
        else:
            d = d + 1
    return f"{n} is prime"
```

Test praštevilskosti – očiten algoritem

Funkcija $n//2$ izračuna $\lfloor n/2 \rfloor$, medtem ko funkcija `n%d` izračuna ostanek pri deljenju n z d.

Algoritem `simple_test` je enostaven, vendar je v najslabšem primeru (ko je n praštevilo) njegova časovna zahtevnost enaka $O(n \log^2 n)$. Torej ta algoritem ni učinkovit.

Časovno zahtevnost algoritma lahko malo izboljšamo s tem, da zanka `while` teče samo do $\sqrt{n}$. Pravilnost algoritma potem sledi iz naslednje opazke: če za delitelj d števila n velja $d \geq \sqrt{n}$, potem za delitelj n/d števila n velja $n/d \leq \sqrt{n}$. Časovna zahtevnost algoritma je potem $O(\sqrt{n} \log^2 n)$, kar še vedno ni učinkovito.

Algoritem bi še vedno pravilno deloval, če bi parameter `d` tekel po množici praštevil, ki ne presegajo števila $\sqrt{n}$. Tedaj bi bila časovna zahtevnost

$$O\left(\pi\left(\sqrt{n}\right) \log^2 n\right) = O\left(\frac{\sqrt{n}}{\log \sqrt{n}} \log^2 n\right) = O\left(\sqrt{n} \log n\right).$$

Test praštevilskosti – očiten algoritem

Funkcija $n//2$ izračuna $\lfloor n/2 \rfloor$, medtem ko funkcija `n%d` izračuna ostanek pri deljenju n z d.

Algoritem `simple_test` je enostaven, vendar je v najslabšem primeru (ko je n praštevilo) njegova časovna zahtevnost enaka $O(n \log^2 n)$. Torej ta algoritem ni učinkovit.

Časovno zahtevnost algoritma lahko malo izboljšamo s tem, da zanka `while` teče samo do $\sqrt{n}$. Pravilnost algoritma potem sledi iz naslednje opazke: če za delitelj d števila n velja $d \geq \sqrt{n}$, potem za delitelj n/d števila n velja $n/d \leq \sqrt{n}$. Časovna zahtevnost algoritma je potem $O(\sqrt{n} \log^2 n)$, kar še vedno ni učinkovito.

Algoritem bi še vedno pravilno deloval, če bi parameter `d` tekel po množici praštevil, ki ne presegajo števila $\sqrt{n}$. Tedaj bi bila časovna zahtevnost

$$O\left(\pi\left(\sqrt{n}\right) \log^2 n\right) = O\left(\frac{\sqrt{n}}{\log \sqrt{n}} \log^2 n\right) = O\left(\sqrt{n} \log n\right).$$

Test praštevilskosti – očiten algoritem

Funkcija $n//2$ izračuna $\lfloor n/2 \rfloor$, medtem ko funkcija `n%d` izračuna ostanek pri deljenju n z d.

Algoritem `simple_test` je enostaven, vendar je v najslabšem primeru (ko je n praštevilo) njegova časovna zahtevnost enaka $O(n \log^2 n)$. Torej ta algoritem ni učinkovit.

Časovno zahtevnost algoritma lahko malo izboljšamo s tem, da zanka `while` teče samo do $\sqrt{n}$. Pravilnost algoritma potem sledi iz naslednje opazke: če za delitelj d števila n velja $d \geq \sqrt{n}$, potem za delitelj n/d števila n velja $n/d \leq \sqrt{n}$. Časovna zahtevnost algoritma je potem $O(\sqrt{n} \log^2 n)$, kar še vedno ni učinkovito.

Algoritem bi še vedno pravilno deloval, če bi parameter `d` tekel po množici praštevil, ki ne presegajo števila $\sqrt{n}$. Tedaj bi bila časovna zahtevnost

$$O\left(\pi\left(\sqrt{n}\right) \log^2 n\right) = O\left(\frac{\sqrt{n}}{\log \sqrt{n}} \log^2 n\right) = O\left(\sqrt{n} \log n\right).$$

Test praštevilskosti – očiten algoritem

Funkcija $n//2$ izračuna $\lfloor n/2 \rfloor$, medtem ko funkcija `n%d` izračuna ostanek pri deljenju n z d.

Algoritem `simple_test` je enostaven, vendar je v najslabšem primeru (ko je n praštevilo) njegova časovna zahtevnost enaka $O(n \log^2 n)$. Torej ta algoritem ni učinkovit.

Časovno zahtevnost algoritma lahko malo izboljšamo s tem, da zanka `while` teče samo do $\sqrt{n}$. Pravilnost algoritma potem sledi iz naslednje opazke: če za delitelj d števila n velja $d \geq \sqrt{n}$, potem za delitelj n/d števila n velja $n/d \leq \sqrt{n}$. Časovna zahtevnost algoritma je potem $O(\sqrt{n} \log^2 n)$, kar še vedno ni učinkovito.

Algoritem bi še vedno pravilno deloval, če bi parameter `d` tekel po množici praštevil, ki ne presegajo števila $\sqrt{n}$. Tedaj bi bila časovna zahtevnost

$$O\left(\pi\left(\sqrt{n}\right) \log^2 n\right) = O\left(\frac{\sqrt{n}}{\log \sqrt{n}} \log^2 n\right) = O\left(\sqrt{n} \log n\right).$$

Fermatov izrek

Pierre de Fermat (1601–1665).

Izrek 6

Naj bo p praštevilo, a naravno število in $p \nmid a$. Potem $a^{p-1} \equiv 1 \pmod{p}$.

Dokaz. Opazimo, da je dovolj dokazati $a^p \equiv a \pmod{p}$. Dokazujemo z metodo matematične indukcije na a. Trditev je očitno pravilna za $a = 1$. Po binomskem izreku imamo

$$(a+1)^p = a^p + \binom{p}{1} a^{p-1} + \cdots + \binom{p}{p-1} a + 1.$$

Po definiciji binomskega simbola imamo tudi

$$k! \binom{p}{k} = p(p-1)\cdots(p-k+1).$$

Naj bo $1 \leq k \leq p-1$. Potem $p \mid k!$ ali $p \mid \binom{p}{k}$. Toda $p \mid k!$ pomeni, da $p \mid j$ za neki $j \in \{1, \ldots, p-1\}$, kar pa je nemogoče. Torej, $p \mid \binom{p}{k}$ za vse $k \in \{1, \ldots, p-1\}$. Sklepamo $(a+1)^p \equiv a^p + 1 \pmod{p}$. Toda $a^p \equiv a \pmod{p}$ po indukcijski predpostavki, zato $(a+1)^p \equiv a + 1 \pmod{p}$. Dokaz je s tem končan.

FERMATOV IZREK

Pierre de Fermat (1601–1665).

Izrek 6

Naj bo p praštevilo, a naravno število in $p \nmid a$. Potem $a^{p-1} \equiv 1 \pmod{p}$.

Dokaz. Opazimo, da je dovolj dokazati $a^p \equiv a \pmod{p}$. Dokazujemo z metodo matematične indukcije na a. Trditev je očitno pravilna za $a = 1$. Po binomskem izreku imamo

$$(a+1)^p = a^p + \binom{p}{1} a^{p-1} + \cdots + \binom{p}{p-1} a + 1.$$

Po definiciji binomskega simbola imamo tudi

$$k! \binom{p}{k} = p(p-1)\cdots(p-k+1).$$

Naj bo $1 \leq k \leq p-1$. Potem $p \mid k!$ ali $p \mid \binom{p}{k}$. Toda $p \mid k!$ pomeni, da $p \mid j$ za neki $j \in \{1, \ldots, p-1\}$, kar pa je nemogoče. Torej, $p \mid \binom{p}{k}$ za vse $k \in \{1, \ldots, p-1\}$. Sklepamo $(a+1)^p \equiv a^p + 1 \pmod{p}$. Toda $a^p \equiv a \pmod{p}$ po indukcijski predpostavki, zato $(a+1)^p \equiv a + 1 \pmod{p}$. Dokaz je s tem končan.

TEST PRAŠTEVILSKOSTI – FERMAT

```
1  from sympy.core.random import _randint
2  from sympy import gcd
3  def fermat_test(n):
4      # n>2 is odd integer
5      a = _randint()(1,n-1)
6      if gcd(a,n) != 1:
7          return f"{n} is composite"
8      else:
9          if pow(a,n-1,n) == 1:
10             return f"{n} is probably prime"
11         else:
12             return f"{n} is composite"
```

Opazke:

1. Uporabljamo Pythonovo knižnico *SymPy* za simbolno matematiko.
2. V vrstici 5 je parameter `a` izbran *(psevdo)naključno* v množici $\{1, 2, \ldots, n-1\}$.
3. V vrstici 6 izraz `gcd(a,n) != 1` pomeni $(a, n) \neq 1$.
4. V vrstici 9 izraz `pow(a,n-1,n) == 1` pomeni $a^{n-1} \mod n = 1$. Na tem mestu algoritem preveri, ali velja $a^{n-1} \equiv 1 \pmod{n}$.

TEST PRAŠTEVILSKOSTI – FERMAT

```
1  from sympy.core.random import _randint
2  from sympy import gcd
3  def fermat_test(n):
4      # n>2 is odd integer
5      a = _randint()(1,n-1)
6      if gcd(a,n) != 1:
7          return f"{n} is composite"
8      else:
9          if pow(a,n-1,n) == 1:
10             return f"{n} is probably prime"
11         else:
12             return f"{n} is composite"
```

Opazke:

1. Uporabljamo Pythonovo knižnico *SymPy* za simbolno matematiko.
2. V vrstici 5 je parameter `a` izbran *(psevdo)naključno* v množici $\{1, 2, \ldots, n-1\}$.
3. V vrstici 6 izraz `gcd(a,n) != 1` pomeni $(a, n) \neq 1$.
4. V vrstici 9 izraz `pow(a,n-1,n) == 1` pomeni $a^{n-1} \mod n = 1$. Na tem mestu algoritem preveri, ali velja $a^{n-1} \equiv 1 \pmod{n}$.

TEST PRAŠTEVILSKOSTI – FERMAT

```
1  from sympy.core.random import _randint
2  from sympy import gcd
3  def fermat_test(n):
4      # n>2 is odd integer
5      a = _randint()(1,n-1)
6      if gcd(a,n) != 1:
7          return f"{n} is composite"
8      else:
9          if pow(a,n-1,n) == 1:
10             return f"{n} is probably prime"
11         else:
12             return f"{n} is composite"
```

Opazke:

1. Uporabljamo Pythonovo knižnico *SymPy* za simbolno matematiko.
2. V vrstici 5 je parameter `a` izbran *(psevdo)naključno* v množici $\{1, 2, \ldots, n-1\}$.
3. V vrstici 6 izraz `gcd(a,n) != 1` pomeni $(a, n) \neq 1$.
4. V vrstici 9 izraz `pow(a,n-1,n) == 1` pomeni $a^{n-1} \mod n = 1$. Na tem mestu algoritem preveri, ali velja $a^{n-1} \equiv 1 \pmod{n}$.

TEST PRAŠTEVILSKOSTI – FERMAT

```
1  from sympy.core.random import _randint
2  from sympy import gcd
3  def fermat_test(n):
4      # n>2 is odd integer
5      a = _randint()(1,n-1)
6      if gcd(a,n) != 1:
7          return f"{n} is composite"
8      else:
9          if pow(a,n-1,n) == 1:
10             return f"{n} is probably prime"
11         else:
12             return f"{n} is composite"
```

Opazke:

1. Uporabljamo Pythonovo knižnico *SymPy* za simbolno matematiko.
2. V vrstici 5 je parameter `a` izbran *(psevdo)naključno* v množici $\{1, 2, \ldots, n-1\}$.
3. V vrstici 6 izraz `gcd(a,n) != 1` pomeni $(a, n) \neq 1$.
4. V vrstici 9 izraz `pow(a,n-1,n) == 1` pomeni $a^{n-1} \mod n = 1$. Na tem mestu algoritem preveri, ali velja $a^{n-1} \equiv 1 \pmod{n}$.

TEST PRAŠTEVILSKOSTI – FERMAT

```
1  from sympy.core.random import _randint
2  from sympy import gcd
3  def fermat_test(n):
4      # n>2 is odd integer
5      a = _randint()(1,n-1)
6      if gcd(a,n) != 1:
7          return f"{n} is composite"
8      else:
9          if pow(a,n-1,n) == 1:
10             return f"{n} is probably prime"
11         else:
12             return f"{n} is composite"
```

Opazke:

1. Uporabljamo Pythonovo knižnico *SymPy* za simbolno matematiko.
2. V vrstici 5 je parameter `a` izbran *(psevdo)naključno* v množici $\{1, 2, \ldots, n-1\}$.
3. V vrstici 6 izraz `gcd(a,n) != 1` pomeni $(a, n) \neq 1$.
4. V vrstici 9 izraz `pow(a,n-1,n) == 1` pomeni $a^{n-1} \mod n = 1$. Na tem mestu algoritem preveri, ali velja $a^{n-1} \equiv 1 \pmod{n}$.

Test praštevilskosti – Fermat

`fermat_test` je primer *randomiziranega algoritma*, saj sloni na naključnih spremenljivkah in lahko tako poda različne odgovore pri enakem vhodu. Pomemben podrazred randomiziranih algoritmov se imenuje *Monte Carlo algoritmi*: izhod je lahko "yes" ali "probably not", in velja naslednje:

1. če algoritem vrne "yes", potem je vhod pozitiven primerek;
2. obstaja realno število $p \in (0, 1]$, da za vsak pozitiven primerek algoritem poda (pravilen) "yes" odgovor z verjetnostjo vsaj p.

Analiza prejšnjega algoritma pokaže naslednje. Če je n praštevilo, potem algoritem `fermat_test` vrne `'n is probably prime'`, kar je posledica Fermatovega izreka. Če pa algoritem vrne `'n is composite'`, potem imamo $(a, n) \neq 1$ ali $a^{n-1} \not\equiv 1 \pmod{n}$ za neki $a \in \{1, \ldots, n-1\}$. Torej je v tem primeru n res sestavljeno število.

Torej, `fermat_test` je randomiziran algoritem, kjer "yes" pomeni `'n is composite'` in pozitiven primerek je sestavljeno število. S tem je algoritem kandidat za Monte Carlo algoritem za odločitveni problem SESTAVLJENA ŠTEVILA.

TEST PRAŠTEVILSKOSTI – FERMAT

`fermat_test` je primer *randomiziranega algoritma*, saj sloni na naključnih spremenljivkah in lahko tako poda različne odgovore pri enakem vhodu. Pomemben podrazred randomiziranih algoritmov se imenuje *Monte Carlo algoritmi*: izhod je lahko "yes" ali "probably not", in velja naslednje:

1. če algoritem vrne "yes", potem je vhod pozitiven primerek;
2. obstaja realno število $p \in (0, 1]$, da za vsak pozitiven primerek algoritem poda (pravilen) "yes" odgovor z verjetnostjo vsaj p.

Analiza prejšnjega algoritma pokaže naslednje. Če je n praštevilo, potem algoritem `fermat_test` vrne `'n is probably prime'`, kar je posledica Fermatovega izreka. Če pa algoritem vrne `'n is composite'`, potem imamo $(a, n) \neq 1$ ali $a^{n-1} \not\equiv 1 \pmod{n}$ za neki $a \in \{1, \ldots, n-1\}$. Torej je v tem primeru n res sestavljeno število.

Torej, `fermat_test` je randomiziran algoritem, kjer "yes" pomeni `'n is composite'` in pozitiven primerek je sestavljeno število. S tem je algoritem kandidat za Monte Carlo algoritem za odločitveni problem SESTAVLJENA ŠTEVILA.

TEST PRAŠTEVILSKOSTI – FERMAT

`fermat_test` je primer *randomiziranega algoritma*, saj sloni na naključnih spremenljivkah in lahko tako poda različne odgovore pri enakem vhodu. Pomemben podrazred randomiziranih algoritmov se imenuje *Monte Carlo algoritmi*: izhod je lahko "yes" ali "probably not", in velja naslednje:

1. če algoritem vrne "yes", potem je vhod pozitiven primerek;
2. obstaja realno število $p \in (0, 1]$, da za vsak pozitiven primerek algoritem poda (pravilen) "yes" odgovor z verjetnostjo vsaj p.

Analiza prejšnjega algoritma pokaže naslednje. Če je n praštevilo, potem algoritem `fermat_test` vrne `'n is probably prime'`, kar je posledica Fermatovega izreka. Če pa algoritem vrne `'n is composite'`, potem imamo $(a, n) \neq 1$ ali $a^{n-1} \not\equiv 1 \pmod{n}$ za neki $a \in \{1, \ldots, n-1\}$. Torej je v tem primeru n res sestavljeno število.

Torej, `fermat_test` je randomiziran algoritem, kjer "yes" pomeni `'n is composite'` in pozitiven primerek je sestavljeno število. S tem je algoritem kandidat za Monte Carlo algoritem za odločitveni problem SESTAVLJENA ŠTEVILA.

Test praštevilskosti – Fermat

Vprašanje: Recimo, da je n sestavljeno liho število. Ali bo algoritem `fermat_test` vrnil `'n is composite'`?

Ne vedno! Recimo, da algoritem izbere $a = 1$ ali $a = n - 1$. Potem je $(a, n) = 1$ in $a^{n-1} \equiv 1 \pmod{n}$, torej bo odgovor `'n is probably prime'`. Dobro, kaj če odstranimo ti dve možnosti, torej naključno izbiramo med naravnimi števili $2, \ldots, n - 2$? Tudi ne vedno, npr.

$$11^{14} = \left(11^2\right)^7 = 121^7 \equiv 1^7 = 1 \pmod{15},$$

toda $15 = 3 \cdot 5$.

Definicija 2

*Naj bo $a \geq 2$ naravno število. Sestavljeno število n je **Frobenijevo psevdopraštevilo k osnovi** a, če velja $a^{n-1} \equiv 1 \pmod{n}$.*

Po prejšnjem primeru je 15 Frobenijevo psevdopraštevilo k osnovi 11. Da se pokazati, da za vsak $a \geq 2$ obstaja neskončno Frobenijevih psevdopraštevil k osnovi a.

TEST PRAŠTEVILSKOSTI – FERMAT

Vprašanje: Recimo, da je n sestavljeno liho število. Ali bo algoritem `fermat_test` vrnil `'n is composite'`?

Ne vedno! Recimo, da algoritem izbere $a = 1$ ali $a = n - 1$. Potem je $(a, n) = 1$ in $a^{n-1} \equiv 1 \pmod{n}$, torej bo odgovor `'n is probably prime'`. Dobro, kaj če odstranimo ti dve možnosti, torej naključno izbiramo med naravnimi števili $2, \ldots, n-2$? Tudi ne vedno, npr.

$$11^{14} = \left(11^2\right)^7 = 121^7 \equiv 1^7 = 1 \pmod{15},$$

toda $15 = 3 \cdot 5$.

Definicija 2

*Naj bo $a \geq 2$ naravno število. Sestavljeno število n je **Frematovo psevdopraštevilo k osnovi** a, če velja $a^{n-1} \equiv 1 \pmod{n}$.*

Po prejšnjem primeru je 15 Frematovo psevdopraštevilo k osnovi 11. Da se pokazati, da za vsak $a \geq 2$ obstaja neskončno Frematovih psevdopraštevil k osnovi a.

TEST PRAŠTEVILSKOSTI – FERMAT

Vprašanje: Recimo, da je n sestavljeno liho število. Ali bo algoritem `fermat_test` vrnil `'n is composite'`?

Ne vedno! Recimo, da algoritem izbere $a = 1$ ali $a = n - 1$. Potem je $(a, n) = 1$ in $a^{n-1} \equiv 1 \pmod{n}$, torej bo odgovor `'n is probably prime'`. Dobro, kaj če odstranimo ti dve možnosti, torej naključno izbiramo med naravnimi števili $2, \ldots, n-2$? Tudi ne vedno, npr.

$$11^{14} = \left(11^2\right)^7 = 121^7 \equiv 1^7 = 1 \pmod{15},$$

toda $15 = 3 \cdot 5$.

Definicija 2

Naj bo $a \geq 2$ naravno število. Sestavljeno število n je ***Frematovo psevdoprašteviloo k osnovi*** *a, če velja $a^{n-1} \equiv 1 \pmod{n}$.*

Po prejšnjem primeru je 15 Frematovo psevdoprašteviloo k osnovi 11. Da se pokazati, da za vsak $a \geq 2$ obstaja neskončno Frematovih psevdopraštevil k osnovi a.

TEST PRAŠTEVILSKOSTI – FERMAT

Vprašanje: Recimo, da je n sestavljeno liho število. Ali bo algoritem `fermat_test` vrnil `'n is composite'`?

Ne vedno! Recimo, da algoritem izbere $a = 1$ ali $a = n - 1$. Potem je $(a, n) = 1$ in $a^{n-1} \equiv 1 \pmod{n}$, torej bo odgovor `'n is probably prime'`. Dobro, kaj če odstranimo ti dve možnosti, torej naključno izbiramo med naravnimi števili $2, \ldots, n-2$? Tudi ne vedno, npr.

$$11^{14} = \left(11^2\right)^7 = 121^7 \equiv 1^7 = 1 \pmod{15},$$

toda $15 = 3 \cdot 5$.

Definicija 2

Naj bo $a \geq 2$ naravno število. Sestavljeno število n je ***Frematovo psevdopraštevilo k osnovi*** *a, če velja $a^{n-1} \equiv 1 \pmod{n}$.*

Po prejšnjem primeru je 15 Frematovo psevdopraštevilo k osnovi 11. Da se pokazati, da za vsak $a \geq 2$ obstaja neskončno Frematovih psevdopraštevil k osnovi a.

TEST PRAŠTEVILSKOSTI – FERMAT

Naj bo

$$\mathscr{A} = \left\{1 \leq a \leq n-1 \colon (a,n) = 1, a^{n-1} \equiv 1 \pmod{n}\right\}.$$

Z uporabo *teorije grup* se da pokazati naslednje.

Trditev 1

Če za dano sestavljeno število n obstaja $a \in \{1, \ldots, n-1\}$ tako da velja $(a,n) = 1$ in $a^{n-1} \not\equiv 1 \pmod{n}$, potem je verjetnost da najdemo tak a vsaj $1/2$.

Torej, če a iz prejšnje trditve vedno obstaja, bo `fermat_test` učinkovit Monte Carlo algoritem (s $p = 1/2$) za odločitveni problem SESTAVLJENA ŠTEVILA.

Robert D. Carmichael (1879–1967).

Obstajajo sestavljena števila, ki so Fermatova psevdoprašteva k poljubni osnovi a. Najmanjše tako število je $561 = 3 \cdot 11 \cdot 17$.

Leta 1994 je bilo dokazano, da je takih števil (*Carmichaelova števila*) neskončno.

TEST PRAŠTEVILSKOSTI – FERMAT

Naj bo

$$\mathscr{A} = \left\{1 \leq a \leq n-1 \colon (a,n) = 1, a^{n-1} \equiv 1 \pmod{n}\right\}.$$

Z uporabo *teorije grup* se da pokazati naslednje.

Trditev 1

Če za dano sestavljeno število n obstaja $a \in \{1, \ldots, n-1\}$ tako da velja $(a,n) = 1$ in $a^{n-1} \not\equiv 1 \pmod{n}$, potem je verjetnost da najdemo tak a vsaj $1/2$.

Torej, če a iz prejšnje trditve vedno obstaja, bo `fermat_test` učinkovit Monte Carlo algoritem (s $p = 1/2$) za odločitveni problem SESTAVLJENA ŠTEVILA.

Robert D. Carmichael (1879–1967).

Obstajajo sestavljena števila, ki so Fermatova psevdoprašteva k poljubni osnovi a. Najmanjše tako število je $561 = 3 \cdot 11 \cdot 17$.

Leta 1994 je bilo dokazano, da je takih števil (*Carmichaelova števila*) neskončno.

TEST PRAŠTEVILSKOSTI – FERMAT

Naj bo

$$\mathscr{A} = \left\{1 \leq a \leq n-1 \colon (a,n) = 1, a^{n-1} \equiv 1 \pmod{n}\right\}.$$

Z uporabo *teorije grup* se da pokazati naslednje.

Trditev 1

Če za dano sestavljeno število n obstaja $a \in \{1, \ldots, n-1\}$ tako da velja $(a,n) = 1$ in $a^{n-1} \not\equiv 1 \pmod{n}$, potem je verjetnost da najdemo tak a vsaj $1/2$.

Torej, če a iz prejšnje trditve vedno obstaja, bo `fermat_test` učinkovit Monte Carlo algoritem (s $p = 1/2$) za odločitveni problem SESTAVLJENA ŠTEVILA.

Robert D. Carmichael (1879–1967).

Obstajajo sestavljena števila, ki so Fermatova psevdoprašteva k poljubni osnovi a. Najmanjše tako število je $561 = 3 \cdot 11 \cdot 17$.

Leta 1994 je bilo dokazano, da je takih števil (*Carmichaelova števila*) neskončno.

TEST PRAŠTEVILSKOSTI – FERMAT

Naj bo

$$\mathscr{A} = \left\{1 \leq a \leq n-1 \colon (a,n) = 1, a^{n-1} \equiv 1 \pmod{n}\right\}.$$

Z uporabo *teorije grup* se da pokazati naslednje.

Trditev 1

Če za dano sestavljeno število n obstaja $a \in \{1, \ldots, n-1\}$ tako da velja $(a,n) = 1$ in $a^{n-1} \not\equiv 1 \pmod{n}$, potem je verjetnost da najdemo tak a vsaj $1/2$.

Torej, če a iz prejšnje trditve vedno obstaja, bo `fermat_test` učinkovit Monte Carlo algoritem (s $p = 1/2$) za odločitveni problem SESTAVLJENA ŠTEVILA.

Robert D. Carmichael (1879–1967).

Obstajajo sestavljena števila, ki so Fermatova psevdoprašteva k poljubni osnovi a. Najmanjše tako število je $561 = 3 \cdot 11 \cdot 17$.

Leta 1994 je bilo dokazano, da je takih števil (*Carmichaelova števila*) neskončno.

TEST PRAŠTEVILSKOSTI – SOLOVAY–STRASSEN

Legendrov in Jacobijev simbol: Naj bo p liho praštevilo, a celo število in $(a,p)=1$. *Legendrov simbol* $(a|p)$ je definiran kot

$$\left(\frac{a}{p}\right) = \begin{cases} 1, & x^2 \equiv a \pmod{p} \text{ je rešljiva,} \\ -1, & x^2 \equiv a \pmod{p} \text{ ni rešljiva.} \end{cases}$$

Dodatno definiramo še $(a|p)=0$ če $p \mid a$. Očitno $(1|p)=1$.

Naj bo $P \geq 3$ liho število s prafaktorizacijo $P = p_1^{\nu_1} \cdots p_r^{\nu_r}$. Potem je *Jacobijev simbol* $(a|P)$ definiran kot

$$\left(\frac{a}{P}\right) = \prod_{i=1}^{r} \left(\frac{a}{p_i}\right)^{\nu_i}.$$

Dodatno definiramo še $(a|1)=1$. Očitno $(1|P)=1$.

Lastnosti:

1. (Eulerjev kriterij) Naj bo p liho praštevilo in $(a,p)=1$. Potem velja $(a|p) \equiv a^{(p-1)/2} \pmod{p}$.
2. Naj bo P liho naravno število. Potem $(ab|P) = (a|P)(b|P)$. Tudi $(a|P) = (b|P)$ za $a \equiv b \pmod{P}$.
3. Naj bo P liho naravno število. Velja

$$\left(\frac{-1}{P}\right) = \begin{cases} 1, & P \equiv 1 \pmod{4}, \\ -1, & P \equiv 3 \pmod{4}, \end{cases} \qquad \left(\frac{2}{P}\right) = \begin{cases} 1, & P \equiv \pm 1 \pmod{8}, \\ -1, & P \equiv \pm 3 \pmod{8}. \end{cases}$$

4. (Zakon kvadratne recipročnosti) Naj bosta P in Q lihi naravni števili. Velja

$$\left(\frac{P}{Q}\right) = \begin{cases} -(Q|P), & P \equiv Q \equiv 3 \pmod{4}, \\ (Q|P), & \text{sicer.} \end{cases}$$

TEST PRAŠTEVILSKOSTI – SOLOVAY–STRASSEN

Legendrov in Jacobijev simbol: Naj bo p liho praštevilo, a celo število in $(a,p)=1$. *Legendrov simbol* $(a|p)$ je definiran kot

$$\left(\frac{a}{p}\right) = \begin{cases} 1, & x^2 \equiv a \pmod{p} \text{ je rešljiva,} \\ -1, & x^2 \equiv a \pmod{p} \text{ ni rešljiva.} \end{cases}$$

Dodatno definiramo še $(a|p)=0$ če $p \mid a$. Očitno $(1|p)=1$.

Naj bo $P \geq 3$ liho število s prafaktorizacijo $P = p_1^{\nu_1} \cdots p_r^{\nu_r}$. Potem je *Jacobijev simbol* $(a|P)$ definiran kot

$$\left(\frac{a}{P}\right) = \prod_{i=1}^{r} \left(\frac{a}{p_i}\right)^{\nu_i}.$$

Dodatno definiramo še $(a|1)=1$. Očitno $(1|P)=1$.

Lastnosti:

1. (Eulerjev kriterij) Naj bo p liho praštevilo in $(a,p)=1$. Potem velja $(a|p) \equiv a^{(p-1)/2} \pmod{p}$.
2. Naj bo P liho naravno število. Potem $(ab|P) = (a|P)(b|P)$. Tudi $(a|P) = (b|P)$ za $a \equiv b \pmod{P}$.
3. Naj bo P liho naravno število. Velja

$$\left(\frac{-1}{P}\right) = \begin{cases} 1, & P \equiv 1 \pmod{4}, \\ -1, & P \equiv 3 \pmod{4}, \end{cases} \qquad \left(\frac{2}{P}\right) = \begin{cases} 1, & P \equiv \pm 1 \pmod{8}, \\ -1, & P \equiv \pm 3 \pmod{8}. \end{cases}$$

4. (Zakon kvadratne recipročnosti) Naj bosta P in Q lihi naravni števili. Velja

$$\left(\frac{P}{Q}\right) = \begin{cases} -(Q|P), & P \equiv Q \equiv 3 \pmod{4}, \\ (Q|P), & \text{sicer.} \end{cases}$$

Test praštevilskosti – Solovay–Strassen

Legendrov in Jacobijev simbol: Naj bo p liho praštevilo, a celo število in $(a,p)=1$. *Legendrov simbol* $(a|p)$ je definiran kot

$$\left(\frac{a}{p}\right) = \begin{cases} 1, & x^2 \equiv a \pmod{p} \text{ je rešljiva,} \\ -1, & x^2 \equiv a \pmod{p} \text{ ni rešljiva.} \end{cases}$$

Dodatno definiramo še $(a|p)=0$ če $p \mid a$. Očitno $(1|p)=1$.
Naj bo $P \geq 3$ liho število s prafaktorizacijo $P = p_1^{\nu_1} \cdots p_r^{\nu_r}$. Potem je *Jacobijev simbol* $(a|P)$ definiran kot

$$\left(\frac{a}{P}\right) = \prod_{i=1}^{r} \left(\frac{a}{p_i}\right)^{\nu_i}.$$

Dodatno definiramo še $(a|1)=1$. Očitno $(1|P)=1$.

Lastnosti:

1. (Eulerjev kriterij) Naj bo p liho praštevilo in $(a,p)=1$. Potem velja $(a|p) \equiv a^{(p-1)/2} \pmod{p}$.
2. Naj bo P liho naravno število. Potem $(ab|P) = (a|P)(b|P)$. Tudi $(a|P) = (b|P)$ za $a \equiv b \pmod{P}$.
3. Naj bo P liho naravno število. Velja

$$\left(\frac{-1}{P}\right) = \begin{cases} 1, & P \equiv 1 \pmod{4}, \\ -1, & P \equiv 3 \pmod{4}, \end{cases} \qquad \left(\frac{2}{P}\right) = \begin{cases} 1, & P \equiv \pm 1 \pmod{8}, \\ -1, & P \equiv \pm 3 \pmod{8}. \end{cases}$$

4. (Zakon kvadratne recipročnosti) Naj bosta P in Q lihi naravni števili. Velja

$$\left(\frac{P}{Q}\right) = \begin{cases} -(Q|P), & P \equiv Q \equiv 3 \pmod{4}, \\ (Q|P), & \text{sicer.} \end{cases}$$

TEST PRAŠTEVILSKOSTI – SOLOVAY–STRASSEN

Legendrov in Jacobijev simbol: Naj bo p liho praštevilo, a celo število in $(a,p)=1$. *Legendrov simbol* $(a|p)$ je definiran kot

$$\left(\frac{a}{p}\right) = \begin{cases} 1, & x^2 \equiv a \pmod{p} \text{ je rešljiva,} \\ -1, & x^2 \equiv a \pmod{p} \text{ ni rešljiva.} \end{cases}$$

Dodatno definiramo še $(a|p)=0$ če $p \mid a$. Očitno $(1|p)=1$.
Naj bo $P \geq 3$ liho število s prafaktorizacijo $P = p_1^{\nu_1} \cdots p_r^{\nu_r}$. Potem je *Jacobijev simbol* $(a|P)$ definiran kot

$$\left(\frac{a}{P}\right) = \prod_{i=1}^{r} \left(\frac{a}{p_i}\right)^{\nu_i}.$$

Dodatno definiramo še $(a|1)=1$. Očitno $(1|P)=1$.

Lastnosti:

1. (Eulerjev kriterij) Naj bo p liho praštevilo in $(a,p)=1$. Potem velja $(a|p) \equiv a^{(p-1)/2} \pmod{p}$.
2. Naj bo P liho naravno število. Potem $(ab|P) = (a|P)(b|P)$. Tudi $(a|P) = (b|P)$ za $a \equiv b \pmod{P}$.
3. Naj bo P liho naravno število. Velja

$$\left(\frac{-1}{P}\right) = \begin{cases} 1, & P \equiv 1 \pmod{4}, \\ -1, & P \equiv 3 \pmod{4}, \end{cases} \qquad \left(\frac{2}{P}\right) = \begin{cases} 1, & P \equiv \pm 1 \pmod{8}, \\ -1, & P \equiv \pm 3 \pmod{8}. \end{cases}$$

4. (Zakon kvadratne recipročnosti) Naj bosta P in Q lihi naravni števili. Velja

$$\left(\frac{P}{Q}\right) = \begin{cases} -(Q|P), & P \equiv Q \equiv 3 \pmod{4}, \\ (Q|P), & \text{sicer.} \end{cases}$$

TEST PRAŠTEVILSKOSTI – SOLOVAY–STRASSEN

Legendrov in Jacobijev simbol: Naj bo p liho praštevilo, a celo število in $(a,p)=1$. *Legendrov simbol* $(a|p)$ je definiran kot

$$\left(\frac{a}{p}\right) = \begin{cases} 1, & x^2 \equiv a \pmod{p} \text{ je rešljiva,} \\ -1, & x^2 \equiv a \pmod{p} \text{ ni rešljiva.} \end{cases}$$

Dodatno definiramo še $(a|p)=0$ če $p \mid a$. Očitno $(1|p)=1$.
Naj bo $P \geq 3$ liho število s prafaktorizacijo $P = p_1^{\nu_1} \cdots p_r^{\nu_r}$. Potem je *Jacobijev simbol* $(a|P)$ definiran kot

$$\left(\frac{a}{P}\right) = \prod_{i=1}^{r} \left(\frac{a}{p_i}\right)^{\nu_i}.$$

Dodatno definiramo še $(a|1)=1$. Očitno $(1|P)=1$.
Lastnosti:

1. (Eulerjev kriterij) Naj bo p liho praštevilo in $(a,p)=1$. Potem velja $(a|p) \equiv a^{(p-1)/2} \pmod{p}$.
2. Naj bo P liho naravno število. Potem $(ab|P) = (a|P)(b|P)$. Tudi $(a|P) = (b|P)$ za $a \equiv b \pmod{P}$.
3. Naj bo P liho naravno število. Velja

$$\left(\frac{-1}{P}\right) = \begin{cases} 1, & P \equiv 1 \pmod{4}, \\ -1, & P \equiv 3 \pmod{4}, \end{cases} \qquad \left(\frac{2}{P}\right) = \begin{cases} 1, & P \equiv \pm 1 \pmod{8}, \\ -1, & P \equiv \pm 3 \pmod{8}. \end{cases}$$

4. (Zakon kvadratne recipročnosti) Naj bosta P in Q lihi naravni števili. Velja

$$\left(\frac{P}{Q}\right) = \begin{cases} -(Q|P), & P \equiv Q \equiv 3 \pmod{4}, \\ (Q|P), & \text{sicer.} \end{cases}$$

TEST PRAŠTEVILSKOSTI – SOLOVAY–STRASSEN

Legendrov in Jacobijev simbol: Naj bo p liho praštevilo, a celo število in $(a,p)=1$. *Legendrov simbol* $(a|p)$ je definiran kot

$$\left(\frac{a}{p}\right) = \begin{cases} 1, & x^2 \equiv a \pmod{p} \text{ je rešljiva,} \\ -1, & x^2 \equiv a \pmod{p} \text{ ni rešljiva.} \end{cases}$$

Dodatno definiramo še $(a|p)=0$ če $p \mid a$. Očitno $(1|p)=1$.
Naj bo $P \geq 3$ liho število s prafaktorizacijo $P = p_1^{\nu_1} \cdots p_r^{\nu_r}$. Potem je *Jacobijev simbol* $(a|P)$ definiran kot

$$\left(\frac{a}{P}\right) = \prod_{i=1}^{r} \left(\frac{a}{p_i}\right)^{\nu_i}.$$

Dodatno definiramo še $(a|1)=1$. Očitno $(1|P)=1$.
Lastnosti:

1. (Eulerjev kriterij) Naj bo p liho praštevilo in $(a,p)=1$. Potem velja $(a|p) \equiv a^{(p-1)/2} \pmod{p}$.
2. Naj bo P liho naravno število. Potem $(ab|P) = (a|P)(b|P)$. Tudi $(a|P) = (b|P)$ za $a \equiv b \pmod{P}$.
3. Naj bo P liho naravno število. Velja

$$\left(\frac{-1}{P}\right) = \begin{cases} 1, & P \equiv 1 \pmod{4}, \\ -1, & P \equiv 3 \pmod{4}, \end{cases} \qquad \left(\frac{2}{P}\right) = \begin{cases} 1, & P \equiv \pm 1 \pmod{8}, \\ -1, & P \equiv \pm 3 \pmod{8}. \end{cases}$$

4. (Zakon kvadratne recipročnosti) Naj bosta P in Q lihi naravni števili. Velja

$$\left(\frac{P}{Q}\right) = \begin{cases} -(Q|P), & P \equiv Q \equiv 3 \pmod{4}, \\ (Q|P), & \text{sicer.} \end{cases}$$

TEST PRAŠTEVILSKOSTI – SOLOVAY–STRASSEN

Legendrov in Jacobijev simbol: Naj bo p liho praštevilo, a celo število in $(a,p)=1$. *Legendrov simbol* $(a|p)$ je definiran kot

$$\left(\frac{a}{p}\right) = \begin{cases} 1, & x^2 \equiv a \pmod{p} \text{ je rešljiva,} \\ -1, & x^2 \equiv a \pmod{p} \text{ ni rešljiva.} \end{cases}$$

Dodatno definiramo še $(a|p)=0$ če $p \mid a$. Očitno $(1|p)=1$.
Naj bo $P \geq 3$ liho število s prafaktorizacijo $P = p_1^{\nu_1} \cdots p_r^{\nu_r}$. Potem je *Jacobijev simbol* $(a|P)$ definiran kot

$$\left(\frac{a}{P}\right) = \prod_{i=1}^{r} \left(\frac{a}{p_i}\right)^{\nu_i}.$$

Dodatno definiramo še $(a|1)=1$. Očitno $(1|P)=1$.
Lastnosti:

1. (Eulerjev kriterij) Naj bo p liho praštevilo in $(a,p)=1$. Potem velja $(a|p) \equiv a^{(p-1)/2} \pmod{p}$.
2. Naj bo P liho naravno število. Potem $(ab|P) = (a|P)(b|P)$. Tudi $(a|P) = (b|P)$ za $a \equiv b \pmod{P}$.
3. Naj bo P liho naravno število. Velja

$$\left(\frac{-1}{P}\right) = \begin{cases} 1, & P \equiv 1 \pmod{4}, \\ -1, & P \equiv 3 \pmod{4}, \end{cases} \qquad \left(\frac{2}{P}\right) = \begin{cases} 1, & P \equiv \pm 1 \pmod{8}, \\ -1, & P \equiv \pm 3 \pmod{8}. \end{cases}$$

4. (Zakon kvadratne recipročnosti) Naj bosta P in Q lihi naravni števili. Velja

$$\left(\frac{P}{Q}\right) = \begin{cases} -(Q|P), & P \equiv Q \equiv 3 \pmod{4}, \\ (Q|P), & \text{sicer.} \end{cases}$$

TEST PRAŠTEVILSKOSTI – SOLOVAY–STRASSEN

Legendrov in Jacobijev simbol: Naj bo p liho praštevilo, a celo število in $(a,p)=1$. *Legendrov simbol* $(a|p)$ je definiran kot

$$\left(\frac{a}{p}\right) = \begin{cases} 1, & x^2 \equiv a \pmod{p} \text{ je rešljiva,} \\ -1, & x^2 \equiv a \pmod{p} \text{ ni rešljiva.} \end{cases}$$

Dodatno definiramo še $(a|p)=0$ če $p \mid a$. Očitno $(1|p)=1$.
Naj bo $P \geq 3$ liho število s prafaktorizacijo $P = p_1^{\nu_1} \cdots p_r^{\nu_r}$. Potem je *Jacobijev simbol* $(a|P)$ definiran kot

$$\left(\frac{a}{P}\right) = \prod_{i=1}^{r} \left(\frac{a}{p_i}\right)^{\nu_i}.$$

Dodatno definiramo še $(a|1)=1$. Očitno $(1|P)=1$.
Lastnosti:

1. (Eulerjev kriterij) Naj bo p liho praštevilo in $(a,p)=1$. Potem velja $(a|p) \equiv a^{(p-1)/2} \pmod{p}$.
2. Naj bo P liho naravno število. Potem $(ab|P) = (a|P)(b|P)$. Tudi $(a|P) = (b|P)$ za $a \equiv b \pmod{P}$.
3. Naj bo P liho naravno število. Velja

$$\left(\frac{-1}{P}\right) = \begin{cases} 1, & P \equiv 1 \pmod{4}, \\ -1, & P \equiv 3 \pmod{4}, \end{cases} \qquad \left(\frac{2}{P}\right) = \begin{cases} 1, & P \equiv \pm 1 \pmod{8}, \\ -1, & P \equiv \pm 3 \pmod{8}. \end{cases}$$

4. (Zakon kvadratne recipročnosti) Naj bosta P in Q lihi naravni števili. Velja

$$\left(\frac{P}{Q}\right) = \begin{cases} -(Q|P), & P \equiv Q \equiv 3 \pmod{4}, \\ (Q|P), & \text{sicer.} \end{cases}$$

TEST PRAŠTEVILSKOSTI – SOLOVAY–STRASSEN

Legendrov in Jacobijev simbol: Naj bo p liho praštevilo, a celo število in $(a,p)=1$. *Legendrov simbol* $(a|p)$ je definiran kot

$$\left(\frac{a}{p}\right) = \begin{cases} 1, & x^2 \equiv a \pmod{p} \text{ je rešljiva,} \\ -1, & x^2 \equiv a \pmod{p} \text{ ni rešljiva.} \end{cases}$$

Dodatno definiramo še $(a|p)=0$ če $p \mid a$. Očitno $(1|p)=1$.
Naj bo $P \geq 3$ liho število s prafaktorizacijo $P = p_1^{\nu_1} \cdots p_r^{\nu_r}$. Potem je *Jacobijev simbol* $(a|P)$ definiran kot

$$\left(\frac{a}{P}\right) = \prod_{i=1}^{r} \left(\frac{a}{p_i}\right)^{\nu_i}.$$

Dodatno definiramo še $(a|1)=1$. Očitno $(1|P)=1$.
Lastnosti:

1. (Eulerjev kriterij) Naj bo p liho praštevilo in $(a,p)=1$. Potem velja $(a|p) \equiv a^{(p-1)/2} \pmod{p}$.
2. Naj bo P liho naravno število. Potem $(ab|P) = (a|P)(b|P)$. Tudi $(a|P) = (b|P)$ za $a \equiv b \pmod{P}$.
3. Naj bo P liho naravno število. Velja

$$\left(\frac{-1}{P}\right) = \begin{cases} 1, & P \equiv 1 \pmod{4}, \\ -1, & P \equiv 3 \pmod{4}, \end{cases} \qquad \left(\frac{2}{P}\right) = \begin{cases} 1, & P \equiv \pm 1 \pmod{8}, \\ -1, & P \equiv \pm 3 \pmod{8}. \end{cases}$$

4. (Zakon kvadratne recipročnosti) Naj bosta P in Q lihi naravni števili. Velja

$$\left(\frac{P}{Q}\right) = \begin{cases} -(Q|P), & P \equiv Q \equiv 3 \pmod{4}, \\ (Q|P), & \text{sicer.} \end{cases}$$

Test praštevilskosti – Solovay–Strassen

Lastnosti (2), (3) in (4) se lahko uporabijo za izračun $(P|Q)$ v polinomskem času, kjer je P celo število in Q je liho naravno število. Algoritem, ki posnema Evklidov algoritem in uporablja "schoolbook" metode za osnovne aritmetične operacije, ima časovno zahtevnost $O(\log(|P|+2)\log(Q+1))$.

Robert M. Solovay in **Volker Strassen** leta 1977 objavita naslednji algoritem.

```
from sympy.functions.combinatorial.numbers import jacobi_symbol
from sympy.core.random import _randint
def solovay_strassen_test(n):
    # n>2 is odd integer
    a = _randint()(1,n-1)
    j = jacobi_symbol(a,n)
    b = pow(a,(n-1)//2,n)
    if j == 0 or (j-b)%n != 0:
        return f"{n} is composite"
    else:
        return f"{n} is probably prime"
```

Algoritem ima časovno zahtevnost $O(\log^3 n)$.

Test praštevilskosti – Solovay–Strassen

Lastnosti (2), (3) in (4) se lahko uporabijo za izračun $(P|Q)$ v polinomskem času, kjer je P celo število in Q je liho naravno število. Algoritem, ki posnema Evklidov algoritem in uporablja "schoolbook" metode za osnovne aritmetične operacije, ima časovno zahtevnost $O(\log(|P|+2)\log(Q+1))$.

Robert M. Solovay in **Volker Strassen** leta 1977 objavita naslednji algoritem.

```
from sympy.functions.combinatorial.numbers import jacobi_symbol
from sympy.core.random import _randint
def solovay_strassen_test(n):
    # n>2 is odd integer
    a = _randint()(1,n-1)
    j = jacobi_symbol(a,n)
    b = pow(a,(n-1)//2,n)
    if j == 0 or (j-b)%n != 0:
        return f"{n} is composite"
    else:
        return f"{n} is probably prime"
```

Algoritem ima časovno zahtevnost $O(\log^3 n)$.

Test praštevilskosti – Solovay–Strassen

Lastnosti (2), (3) in (4) se lahko uporabijo za izračun $(P|Q)$ v polinomskem času, kjer je P celo število in Q je liho naravno število. Algoritem, ki posnema Evklidov algoritem in uporablja "schoolbook" metode za osnovne aritmetične operacije, ima časovno zahtevnost $O(\log(|P|+2)\log(Q+1))$.

Robert M. Solovay in **Volker Strassen** leta 1977 objavita naslednji algoritem.

```
from sympy.functions.combinatorial.numbers import jacobi_symbol
from sympy.core.random import _randint
def solovay_strassen_test(n):
    # n>2 is odd integer
    a = _randint()(1,n-1)
    j = jacobi_symbol(a,n)
    b = pow(a,(n-1)//2,n)
    if j == 0 or (j-b)%n != 0:
        return f"{n} is composite"
    else:
        return f"{n} is probably prime"
```

Algoritem ima časovno zahtevnost $O(\log^3 n)$.

Test praštevilskosti – Solovay–Strassen

Test Solovay–Strassen je učinkovit Monte Carlo algoritem (s $p = 1/2$) za odločitveni problem SESTAVLJENA ŠTEVILA.

Izrek 7

Naj bo n liho sestavljeno naravno število. Potem obstaja tak $a \in \{1, \ldots, n-1\}$, da velja $(a, n) = 1$ in $(a|n) \not\equiv a^{(n-1)/2} \pmod{n}$.

Posledica 1

Naj bo n liho sestavljeno naravno število. Potem

$$\left|\left\{a \in \mathbb{Z}_n^* \colon a^{(n-1)/2} \equiv \left(\frac{a}{n}\right) \pmod{n}\right\}\right| \leq \frac{1}{2}\varphi(n).$$

Test praštevilskosti – Solovay–Strassen

Test Solovay–Strassen je učinkovit Monte Carlo algoritem (s $p = 1/2$) za odločitveni problem SESTAVLJENA ŠTEVILA.

Izrek 7

Naj bo n liho sestavljeno naravno število. Potem obstaja tak $a \in \{1, \ldots, n-1\}$, da velja $(a, n) = 1$ in $(a|n) \not\equiv a^{(n-1)/2} \pmod{n}$.

Posledica 1

Naj bo n liho sestavljeno naravno število. Potem

$$\left|\left\{a \in \mathbb{Z}_n^* \colon a^{(n-1)/2} \equiv \left(\frac{a}{n}\right) \pmod{n}\right\}\right| \leq \frac{1}{2}\varphi(n).$$

Test praštevilskosti – Miller–Rabin

Gary L. Miller in **Michael O. Rabin** sta (neodvisno) "popravila" Fermatov pristop ter dobila učinkovit Monte Carlo algoritem (s $p = 3/4$) za odločitveni problem SESTAVLJENA ŠTEVILA.

Izrek 8 (Miller)

Naj bo p liho praštevilo. Pišimo $p - 1 = d \cdot 2^l$, kjer je $d \in \mathbb{N}$, $l \in \mathbb{N}$ in $2 \nmid d$. Če je a tako naravno število, da velja $p \nmid a$, potem:

1. $a^d \equiv 1 \pmod{p}$, *ali*
2. $a^{d \cdot 2^i} \equiv -1 \pmod{p}$ *za neki* $i \in \{0, 1, \ldots, l-1\}$.

Izrek 9 (Rabin)

Naj bo n liho sestavljeno naravno število. Pišimo $n - 1 = d \cdot 2^l$, kjer je $d \in \mathbb{N}$, $l \in \mathbb{N}$ in $2 \nmid d$. Potem imamo

$$\left| \left\{ a \in \{1, \ldots, n-1\} : a^d \not\equiv 1 \pmod{n} \wedge a^{d \cdot 2^i} \not\equiv -1 \pmod{n} \, \forall i \in \{0, \ldots, l-1\} \right\} \right| \geq \frac{3}{4}(n-1).$$

Test praštevilskosti – Miller–Rabin

Gary L. Miller in **Michael O. Rabin** sta (neodvisno) "popravila" Fermatov pristop ter dobila učinkovit Monte Carlo algoritem (s $p = 3/4$) za odločitveni problem SESTAVLJENA ŠTEVILA.

Izrek 8 (Miller)

Naj bo p liho praštevilo. Pišimo $p - 1 = d \cdot 2^l$, kjer je $d \in \mathbb{N}$, $l \in \mathbb{N}$ in $2 \nmid d$. Če je a tako naravno število, da velja $p \nmid a$, potem:

1. $a^d \equiv 1 \pmod{p}$, *ali*
2. $a^{d \cdot 2^i} \equiv -1 \pmod{p}$ *za neki* $i \in \{0, 1, \ldots, l-1\}$.

Izrek 9 (Rabin)

Naj bo n liho sestavljeno naravno število. Pišimo $n - 1 = d \cdot 2^l$, kjer je $d \in \mathbb{N}$, $l \in \mathbb{N}$ in $2 \nmid d$. Potem imamo

$$\left|\left\{a \in \{1, \ldots, n-1\} : a^d \not\equiv 1 \pmod{n} \wedge a^{d \cdot 2^i} \not\equiv -1 \pmod{n} \, \forall i \in \{0, \ldots, l-1\}\right\}\right| \geq \frac{3}{4}(n-1).$$

Test praštevilskosti – Miller–Rabin

Gary L. Miller in **Michael O. Rabin** sta (neodvisno) "popravila" Fermatov pristop ter dobila učinkovit Monte Carlo algoritem (s $p = 3/4$) za odločitveni problem SESTAVLJENA ŠTEVILA.

Izrek 8 (Miller)

Naj bo p liho praštevilo. Pišimo $p - 1 = d \cdot 2^l$, kjer je $d \in \mathbb{N}$, $l \in \mathbb{N}$ in $2 \nmid d$. Če je a tako naravno število, da velja $p \nmid a$, potem:

1. $a^d \equiv 1 \pmod{p}$, *ali*
2. $a^{d \cdot 2^i} \equiv -1 \pmod{p}$ *za neki* $i \in \{0, 1, \ldots, l-1\}$.

Izrek 9 (Rabin)

Naj bo n liho sestavljeno naravno število. Pišimo $n - 1 = d \cdot 2^l$, kjer je $d \in \mathbb{N}$, $l \in \mathbb{N}$ in $2 \nmid d$. Potem imamo

$$\left|\left\{a \in \{1, \ldots, n-1\} : a^d \not\equiv 1 \pmod{n} \wedge a^{d \cdot 2^i} \not\equiv -1 \pmod{n} \, \forall i \in \{0, \ldots, l-1\}\right\}\right| \geq \frac{3}{4}(n-1).$$

Test praštevilskosti – Miller–Rabin

```python
from sympy.core.random import _randint
from sympy import gcd
def miller_rabin_test(n):
    # n>2 is odd integer
    d = n-1
    l = 0
    while d%2 == 0:
        l = l+1
        d = d//2
    a = _randint()(1,n-1)
    if gcd(a,n) != 1:
        return f"{n} is composite"
    else:
        b = pow(a,d,n)
        if b == 1:
            return f"{n} is probably prime"
        else:
            for i in range(l):
                if (b+1)%n == 0:
                    return f"{n} is probably prime"
                else:
                    b = pow(b,2,n)
            return f"{n} is composite"
```

Algoritem ima časovno zahtevnost $O(\log^3 n)$ in je v praksi hitrejši kot Solovay–Strassen.

Test praštevilskosti – Miller–Rabin

```python
from sympy.core.random import _randint
from sympy import gcd
def miller_rabin_test(n):
    # n>2 is odd integer
    d = n-1
    l = 0
    while d%2 == 0:
        l = l+1
        d = d//2
    a = _randint()(1,n-1)
    if gcd(a,n) != 1:
        return f"{n} is composite"
    else:
        b = pow(a,d,n)
        if b == 1:
            return f"{n} is probably prime"
        else:
            for i in range(l):
                if (b+1)%n == 0:
                    return f"{n} is probably prime"
                else:
                    b = pow(b,2,n)
            return f"{n} is composite"
```

Algoritem ima časovno zahtevnost $O(\log^3 n)$ in je v praksi hitrejši kot Solovay–Strassen.

Test praštevilskosti – Miller–Rabin

Algoritem `miller_rabin_test` lahko večkrat uporabimo na istem lihem številu. Ideja je, da če algoritem vrne `'n is probably prime'` za vsak test, in če je testov zadostno število, potem je verjetnost, da je testno število res praštevilo, blizu 1. Formalno lahko to utemeljimo z verjetnostnim računom.

Naj bo $\mathcal{P}$ dogodek, da je naključno izbrano število (ki ga testiramo) iz podmnožice $(N, M] \cap \mathbb{N}$ naravnih števil praštevilo, in naj bo $\mathcal{MR}_k$ dogodek, da to število opravi test vseh k ciklov. Zanima nas $P(\mathcal{P}|\mathcal{MR}_k)$, to je verjetnost, da je testirano število praštevilo, ko je Miller–Rabinov test že vrnil `'n is probably prime'` vseh k ciklov. *Bayesova formula* nam potem zagotavlja

$$P(\mathcal{P}|\mathcal{MR}_k) \geq 1 - \frac{(M-N)(1-3/4)^k}{\pi(M) - \pi(N)}.$$

Ta izraz lahko dodatno poenostavimo z uporabo ekplicitne oblike praštevilskega izreka.

Test praštevilskosti – Miller–Rabin

Algoritem `miller_rabin_test` lahko večkrat uporabimo na istem lihem številu. Ideja je, da če algoritem vrne `'n is probably prime'` za vsak test, in če je testov zadostno število, potem je verjetnost, da je testno število res praštevilo, blizu 1. Formalno lahko to utemeljimo z verjetnostnim računom.

Naj bo $\mathcal{P}$ dogodek, da je naključno izbrano število (ki ga testiramo) iz podmnožice $(N, M] \cap \mathbb{N}$ naravnih števil praštevilo, in naj bo $\mathcal{MR}_k$ dogodek, da to število opravi test vseh k ciklov. Zanima nas $P(\mathcal{P}|\mathcal{MR}_k)$, to je verjetnost, da je testirano število praštevilo, ko je Miller–Rabinov test že vrnil `'n is probably prime'` vseh k ciklov. *Bayesova formula* nam potem zagotavlja

$$P(\mathcal{P}|\mathcal{MR}_k) \geq 1 - \frac{(M-N)(1-3/4)^k}{\pi(M) - \pi(N)}.$$

Ta izraz lahko dodatno poenostavimo z uporabo ekplicitne oblike praštevilskega izreka.

Kaj pa učinkovit deterministični algoritem?

Miller je pokazal, da se da `miller_rabin_test` preoblikovati do učinkovitega deterministčnega algoritma, če le privzamemo pravilnost (še vedno ne dokazane) *posplošene Riemannove domneve*. Ker je splošno prepričanje, da je domneva pravilna, je bilo tudi prepričanje, da učinkovit deterministični algoritem za odločitveni problem PRAŠTEVILA obstaja.

To so leta 2004 dokazali **Manindra Agrawal**, **Neeraj Kayal** in **Nitin Saxena**. Njihov algoritem ima časovno zahtevnost $O(\log^{6+o(1)} n)$ in je v praksi mnogo počasnejši kot Miller–Rabin.

Več o tem si lahko preberete v knjigi *Primality testing for beginners* avtorjev L. Rempe-Gillen in R. Waldecker, ki je izšla pri založbi American Mathematical Society leta 2014.

Kaj pa učinkovit deterministični algoritem?

Miller je pokazal, da se da `miller_rabin_test` preoblikovati do učinkovitega deterističnega algoritma, če le privzamemo pravilnost (še vedno ne dokazane) *posplošene Riemannove domneve*. Ker je splošno prepričanje, da je domneva pravilna, je bilo tudi prepričanje, da učinkovit deterministični algoritem za odločitveni problem PRAŠTEVILA obstaja.

To so leta 2004 dokazali **Manindra Agrawal**, **Neeraj Kayal** in **Nitin Saxena**. Njihov algoritem ima časovno zahtevnost $O(\log^{6+o(1)} n)$ in je v praksi mnogo počasnejši kot Miller–Rabin.

Več o tem si lahko preberete v knjigi *Primality testing for beginners* avtorjev L. Rempe-Gillen in R. Waldecker, ki je izšla pri založbi American Mathematical Society leta 2014.

Kaj pa učinkovit deterministični algoritem?

Miller je pokazal, da se da `miller_rabin_test` preoblikovati do učinkovitega determinističnega algoritma, če le privzamemo pravilnost (še vedno ne dokazane) *posplošene Riemannove domneve*. Ker je splošno prepričanje, da je domneva pravilna, je bilo tudi prepričanje, da učinkovit deterministični algoritem za odločitveni problem PRAŠTEVILA obstaja.

To so leta 2004 dokazali **Manindra Agrawal**, **Neeraj Kayal** in **Nitin Saxena**. Njihov algoritem ima časovno zahtevnost $O(\log^{6+o(1)} n)$ in je v praksi mnogo počasnejši kot Miller–Rabin.

Več o tem si lahko preberete v knjigi *Primality testing for beginners* avtorjev L. Rempe-Gillen in R. Waldecker, ki je izšla pri založbi American Mathematical Society leta 2014.

Hvala za pozornost!