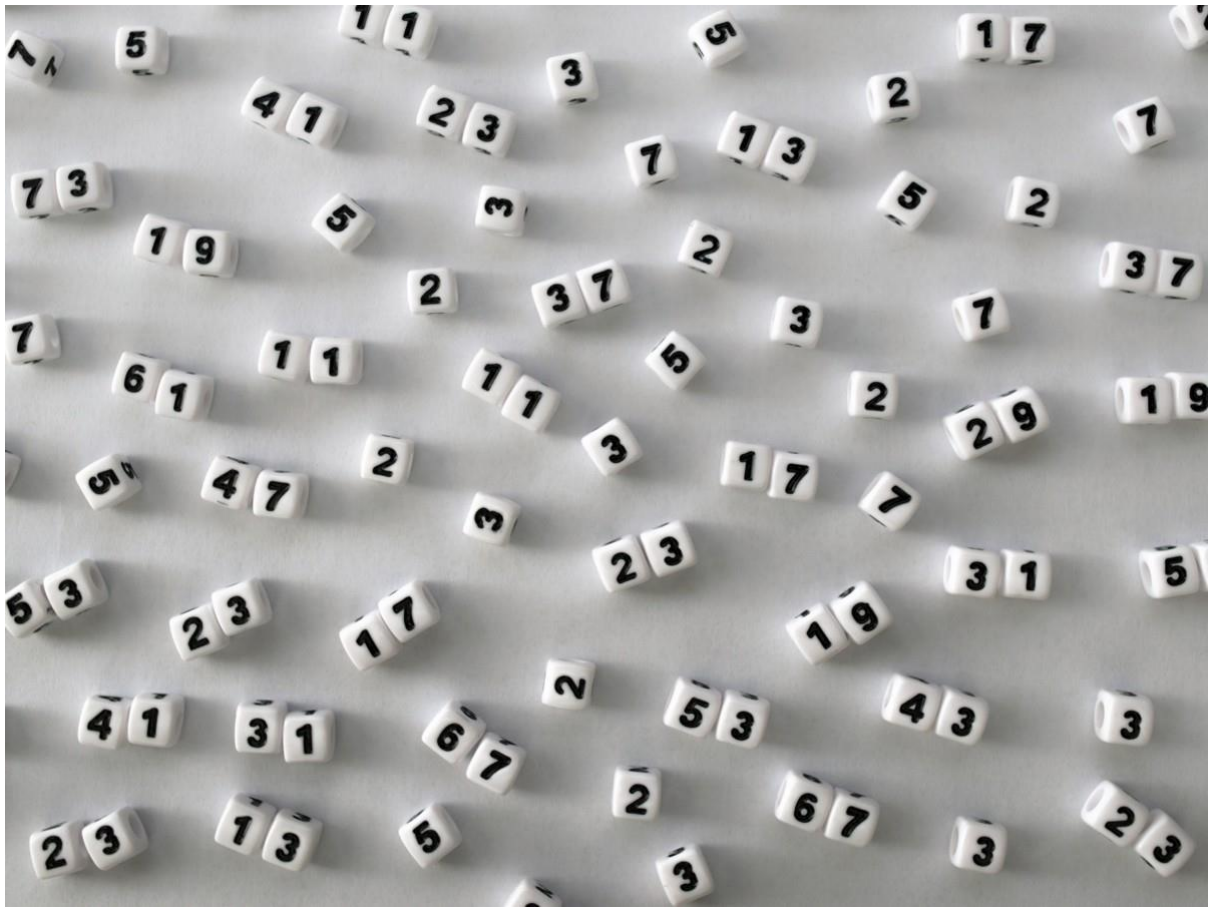
Berechnen von Primzahlen

GRA WS 22/23
Gruppe 151

Aaron Schulze
Ignat Kharlamov
Kerem Çolak



[1] Robert Brook/Getty Images

Einführung

- Was ist eine Primzahl?
- Satz des Euklid [2]:

  P = $\{2, 3, 5, 7, 11, \dots, p_n\}$ endliche Menge aller Primzahlen

  $p_{n+1} = (2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot \dots \cdot p_n) + 1$

- Fundamentalsatz der Arithmetik [3]:

  Beispiel Faktorisierung: $126 = 2^1 \cdot 3^2 \cdot 7^1$
- Wofür Primzahlen?

  RSA und DH, Schlüsselgenerierung, Zufallszahlen, Hashverfahren

Primzahlen berechnen

- Primzahltests, gegeben eine Zahl *n*
- Algorithmen zur Entwicklung von Primzahlenfolgen
- Einfache Methode der Probedivision:

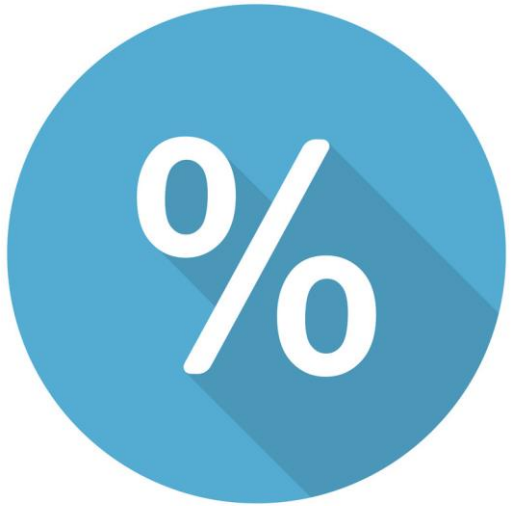
  Beispiel $n = 17$

  $(17\ mod\ 2) = \ \ 1$

  $(17\ mod\ 3) = \ \ 2$

  …

  $(17\ mod\ 16) = 1$

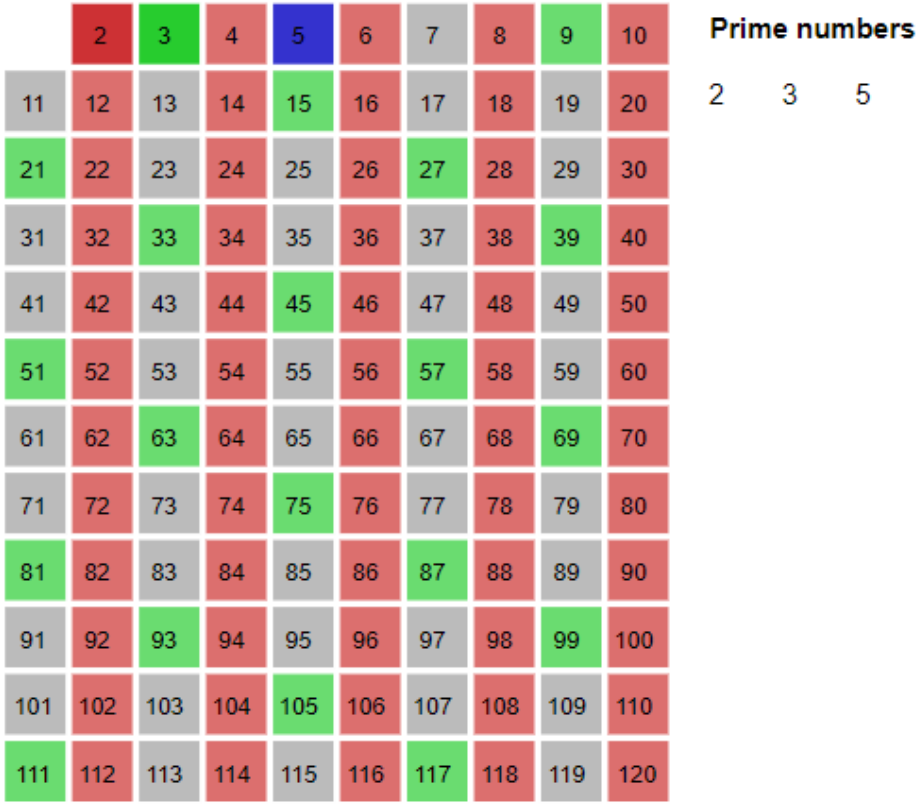


[4]

- Optimierungen: ungerade Zahlen, vorberechnete Primzahlen, Grenze $\sqrt{n}$

Das Sieb des Eratosthenes

- Praktische Lösung für das Ermitteln aller Primzahlen zwischen 2 und *n*
- Markierung zur Unterscheidung zwischen Primzahlen und zusammengesetzte Zahlen
- Anfang: alle Zahlen sind potenzielle Primzahlen
- Streichen aller Vielfachen einer gefundenen Primzahl
- Optimierungen möglich
- Nachteil: Speicherbedarf



[5]

Anwendung von SIMD

- Verwendung: Array von 128-Bit-Vektoren

  → Speicheroptimierung
- Generierung einer Bit-Maske für jeden Vektor

  für die Markierung der Vielfachen
- Logischer Shift um p Stellen und Negation

[2,3,4,5,6,7,…,130] - Primkandidaten

↓

[1,1,1,1,1,1,…,1] - 128-Bit Siebvektor

[1,0,0,0,0,0,...,0] - Offsetvektor, p=2

[1,1,0,1,1,1,...,1] - Offsetvektor, erste Vielfache

Anwendung von SIMD

- Logische Verundung mit dem Siebvektor
- Aktualisieren der Bit-Maske
- Iterative Eliminierung der Vielfachen

[1,1,1,1,1,1,…,1] - 128-Bit Siebvektor

&

[1,1,0,1,1,1,...,1] - Offsetvektor

=

[1,1,0,1,1,1,…,1] - Ergebnis: Siebvektor

[1,1,0,1,1,1,…,1] - Siebvektor

&

[1,1,1,1,0,1,...,1] - Offsetvektor

=

[1,1,0,1,0,1,…,1] - Ergebnis: Siebvektor

Anwendung von SIMD

- Alle Vielfachen aus Siebvektor eliminiert
- Reset Offset und Shift um eine Stelle nach Rechts
- Verundung: nächste Primzahl p

[2,3,4,5,6,7,…,130] - Primkandidaten

↓

[1,1,0,1,0,1,…,0] - 128-Bit Siebvektor

[1,0,0,0,0,0,...,0] - Reset offset

[0,1,0,0,0,0,...,0] - Shift offset

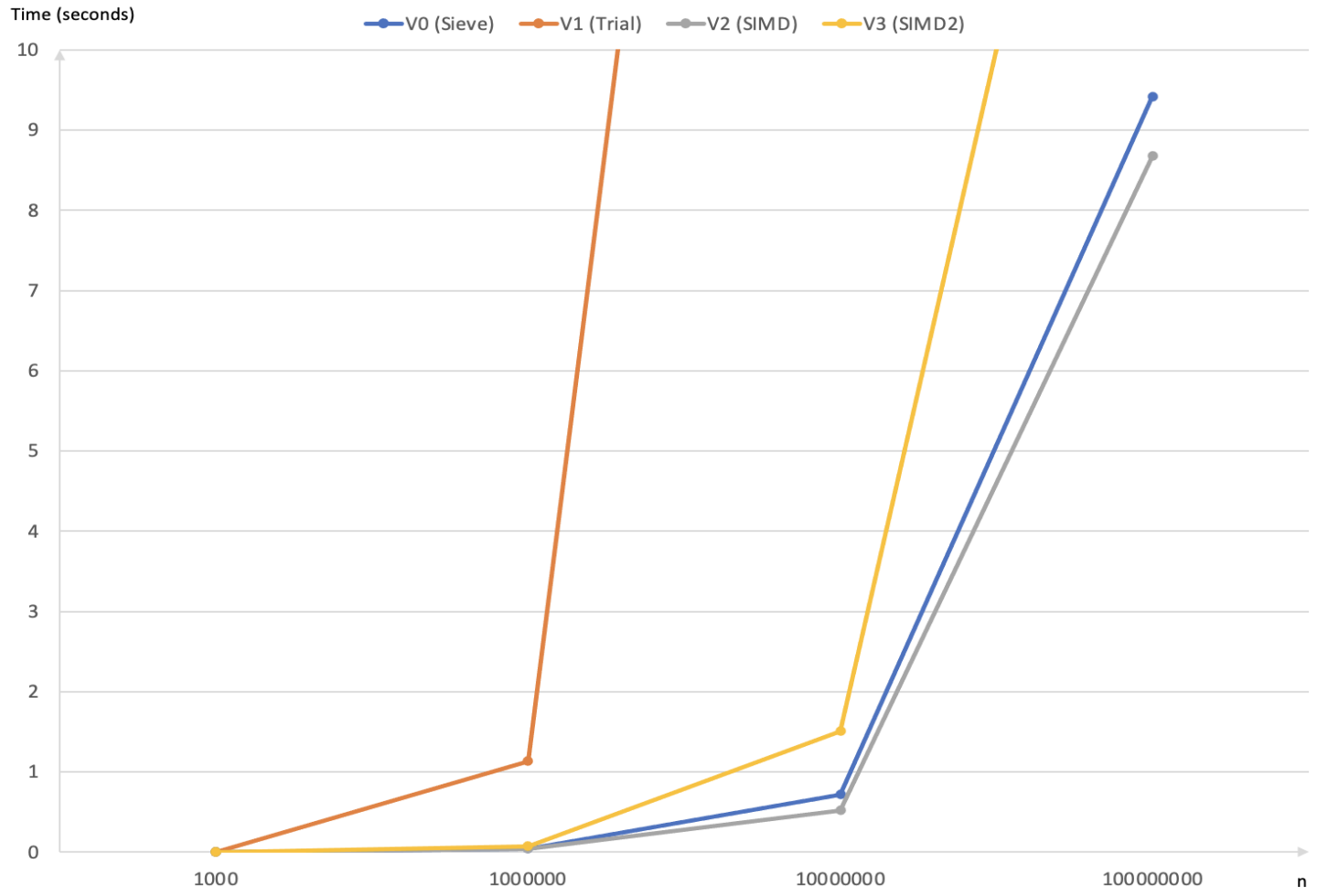
&

[1,1,0,1,0,1,…,0] - 128-Bit Siebvektor

[0,0,0,0,1,0,...,0] - Shift offset um p=3

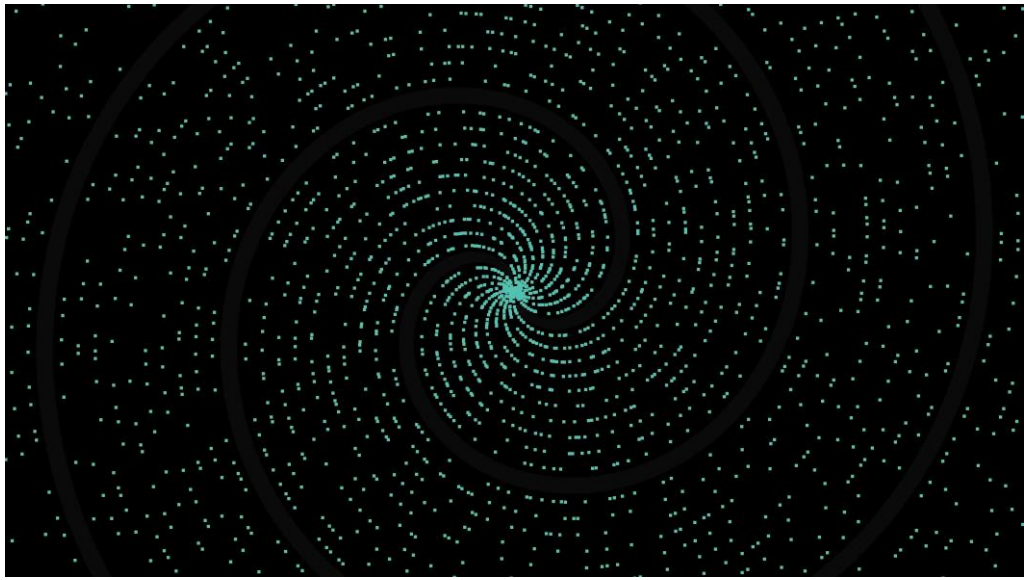
[1,1,1,1,0,1,...,1] - Neg offset

Performanzanalyse



Zusammenfassung

- Unterschiedliche Verfahren und Ansätze
- Keine eindeutige Muster
- Geringere Häufigkeit für große Werte
- Moderne Verfahren basierend auf Bibliotheksfunktionen oder prob. Tests
- Quantencomputer



[6]

Referenzen für die Slide-Inhalte

[1]: https://sciencephotogallery.com/featured/1-prime-numbers-robert-brookscience-photo-library.html

[2]: https://primes.utm.edu/notes/proofs/infinite/euclids.html

[3]: https://www.spektrum.de/lexikon/mathematik/fundamentalsatz-der-arithmetik/3235

[4]: https://www.vectorstock.com/royalty-free-vector/percent-sign-vector-28317607

[5]: https://upload.wikimedia.org/wikipedia/commons/0/0d/Sieve_of_Eratosthenes_animation-en.svg

[6]: https://www.3blue1brown.com/lessons/prime-spirals