

LEHRSTUHL FÜR RECHENARCHITEKTUR UND PARALLELE SYSTEME

# Grundlagenpraktikum: Rechenarchitektur

Gruppe 105 – Abgabe zur Aufgabe A326

Sommersemester 2023

Benedikt Falk

Fabian Strobl

Severin Leitner

$$\sqrt{2}$$

# Gliederung

- Problemstellung
- Funktionsweise
- Schwierigkeiten
- Konzepte
  - Struct „BigNum“
  - Grundrechenarten
  - Schnelle Exponentiation
- Performanceanalyse
- Ausblick

# Problemstellung

- $\sqrt{2}$  auf beliebige Genauigkeit
- berechnen
- Die erste bewiesene irrationale Zahl
- Es gibt unterschiedliche Arten der
- Berechnung
- Rekord liegt bei 10 Billionen Stellen



# Unser Verfahren und Nutzung in anderen Anwendungsgebieten

- Exponentiation der Matrix:  
 $\{\{0,1\},\{1,2\}\}$
- Konvergiert zu  $-1+\sqrt{2}$
- Aufgrund der Skalierung des maximalen Eigenvektors
- Pagerank
- Markov-Chains



# Funktionsweise des Programms

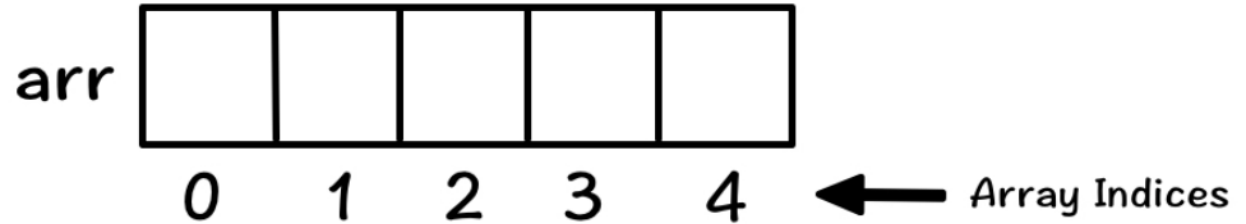
- Genauigkeit der Ausgabe durch den Nutzer bestimmt
- Ausgabe mit unterschiedlichen Implementierungen
- Entweder in hexadezimaler oder dezimaler Darstellung
- Messung der Performance

# Schwierigkeiten

- Benutzung von internen Datenstrukturen (z.B. long oder int) sehr limitiert
- Zahlen werden schnell zu groß
- Performanceprobleme
- Komplexität von eigentlich simplen Rechnungen

# Struct bignum

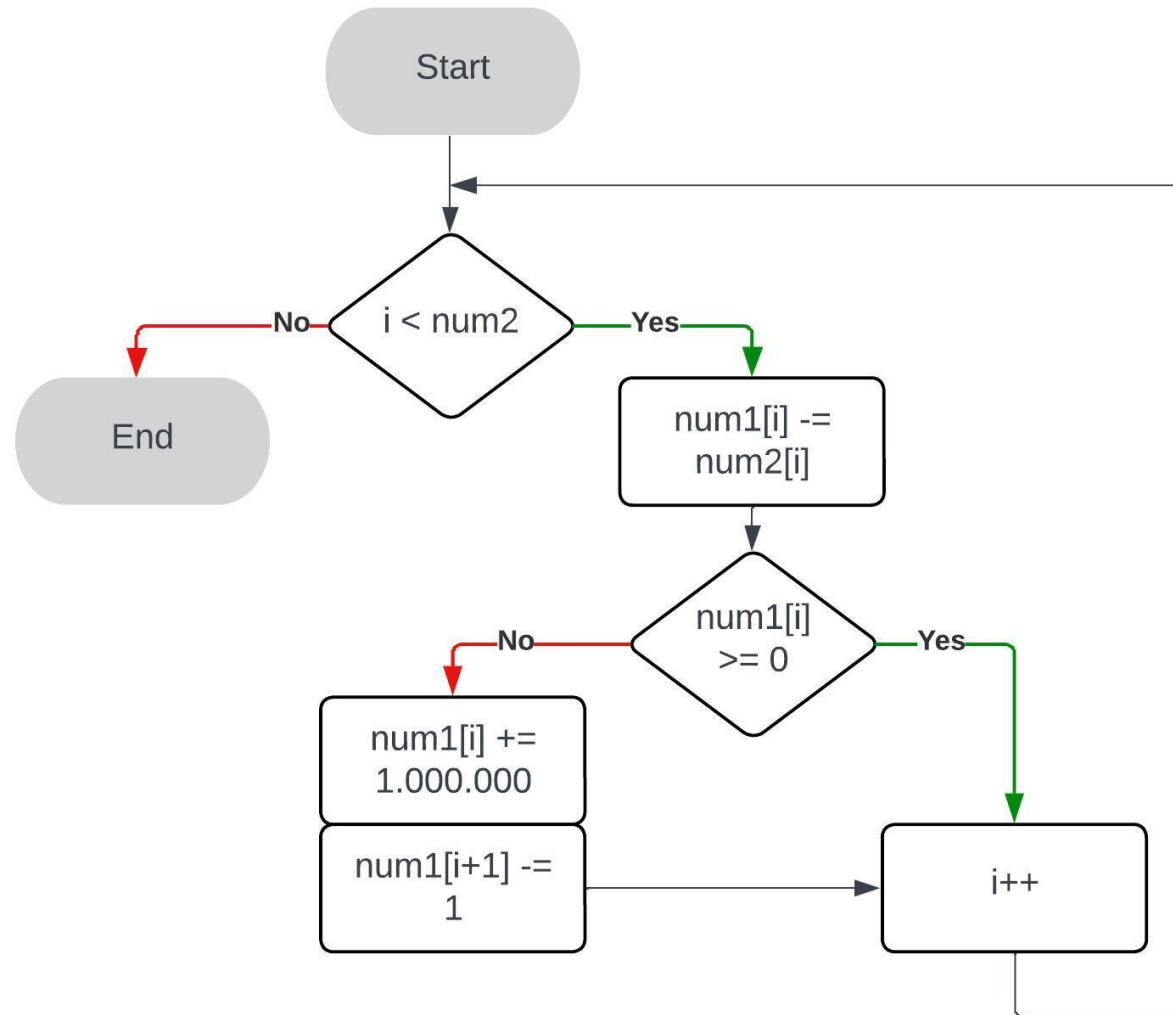
- Setzt sich zusammen aus
  - Einem long Array
  - Der Länge des Arrays
- Pro Array Werte zwischen 0 – 999.999.999
- Speichert im Array 0 die "kleinsten Werte"



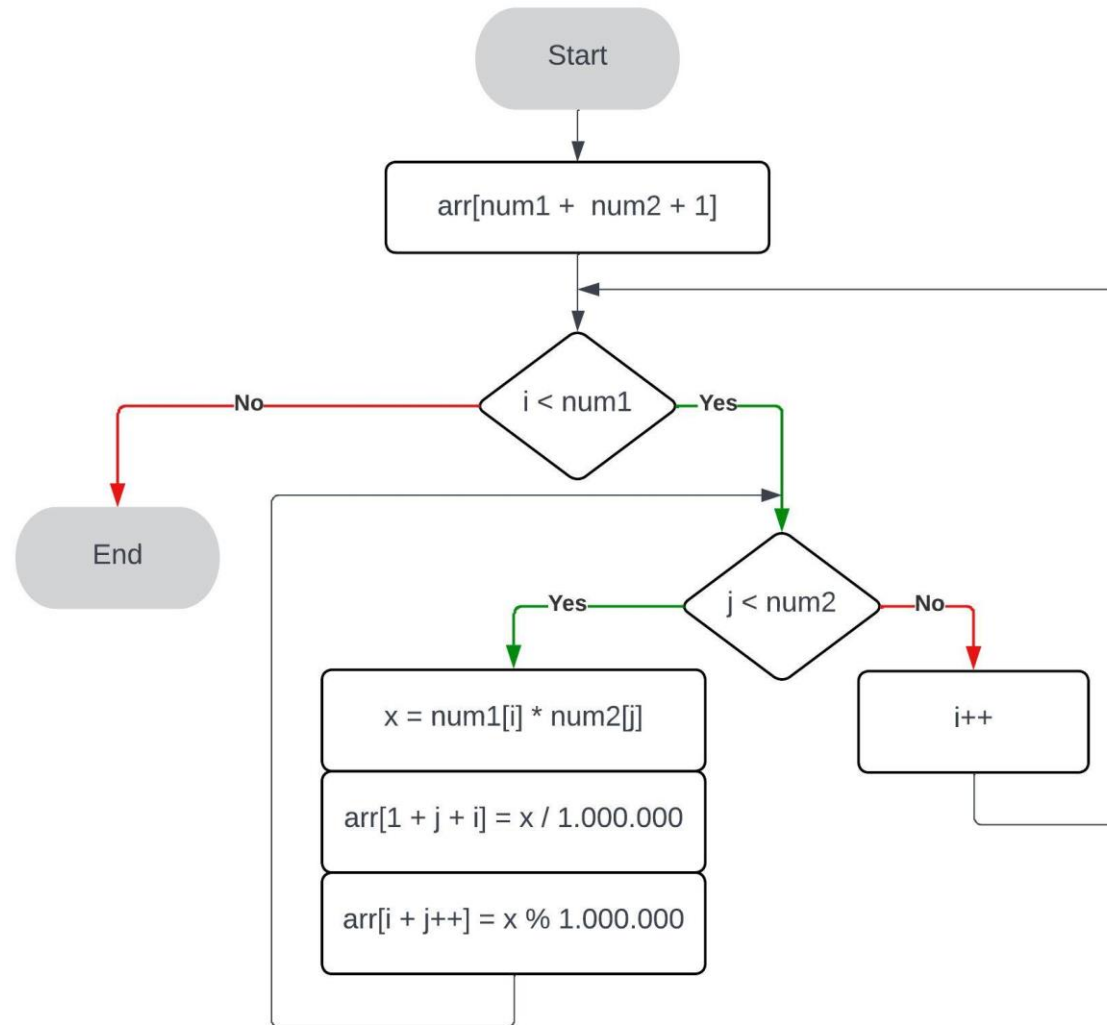




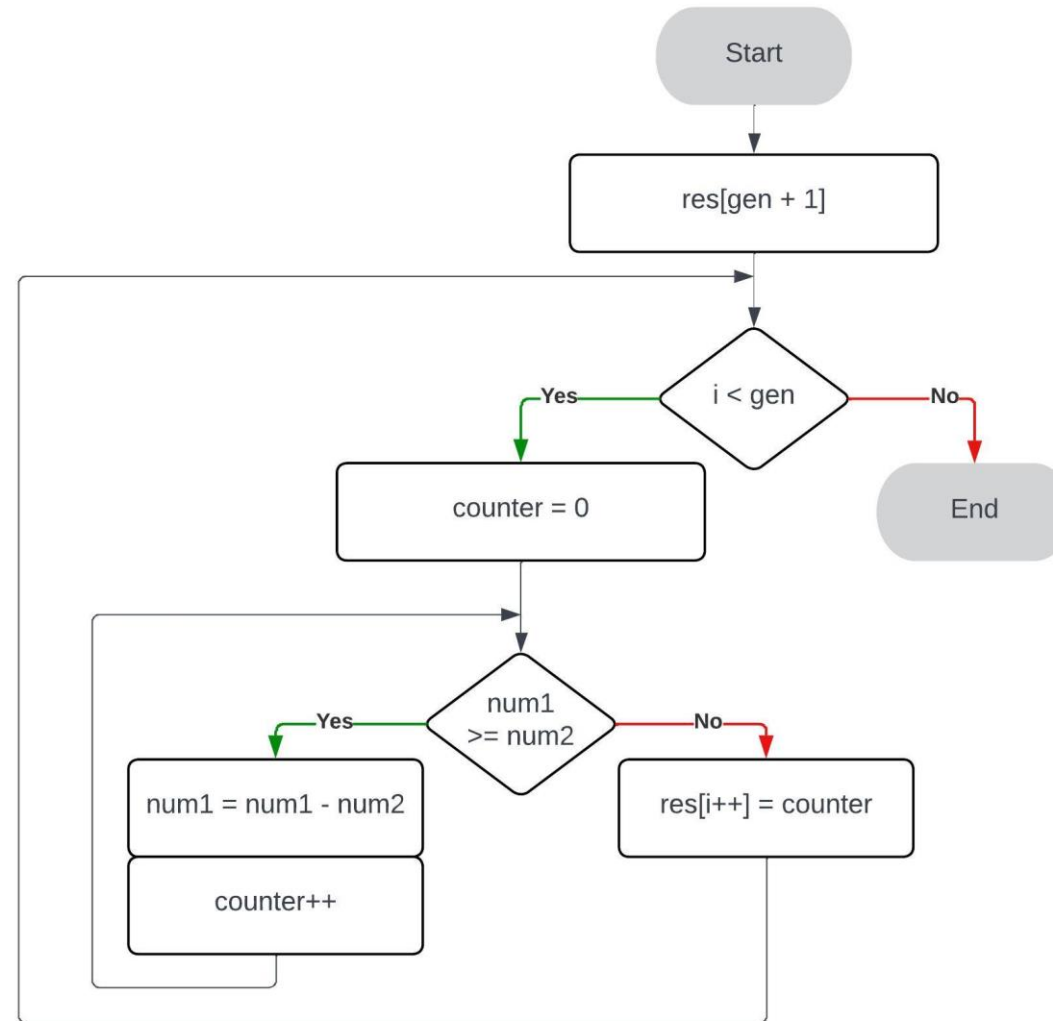
# SUB



# MUL



# DIV



# Schnelle Exponentiation

$$7^{11} \mid 2^4 = 16 \mid 11 = 1011_b \mid res = 1; val = base;$$

$$\text{Bit an Stelle } i := \begin{cases} 1, & res * val \\ 0, & - \end{cases}$$

$$val^2$$

# Schnelle Exponentiation

$i$	$a_i$
0	$7 = 7$
1	$7^2 = 49$
2	$49^2 = 2401$
3	$2401^2 = 5764801$

$$11 = 1011_b$$

$$A = 1 * a_0$$

$$A = 1 * a_0 * a_1$$

$$A = 1 * a_0 * a_1$$

$$A = 1 * a_0 * a_1 * a_3$$

$$A = 1 * 7 * 49 * 5764801$$

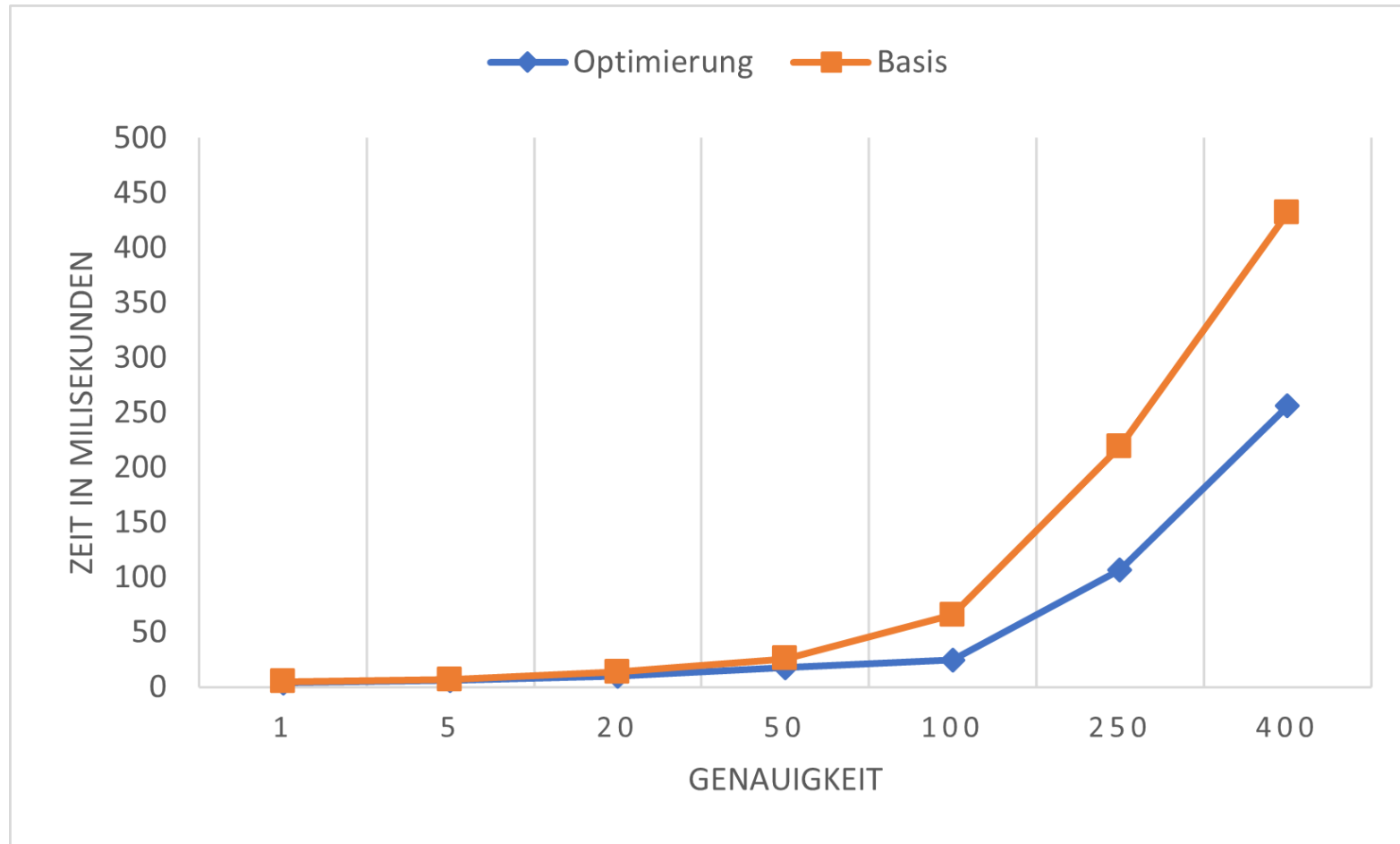
# Schnelle Exponentiation

$$11 = 1011_b$$

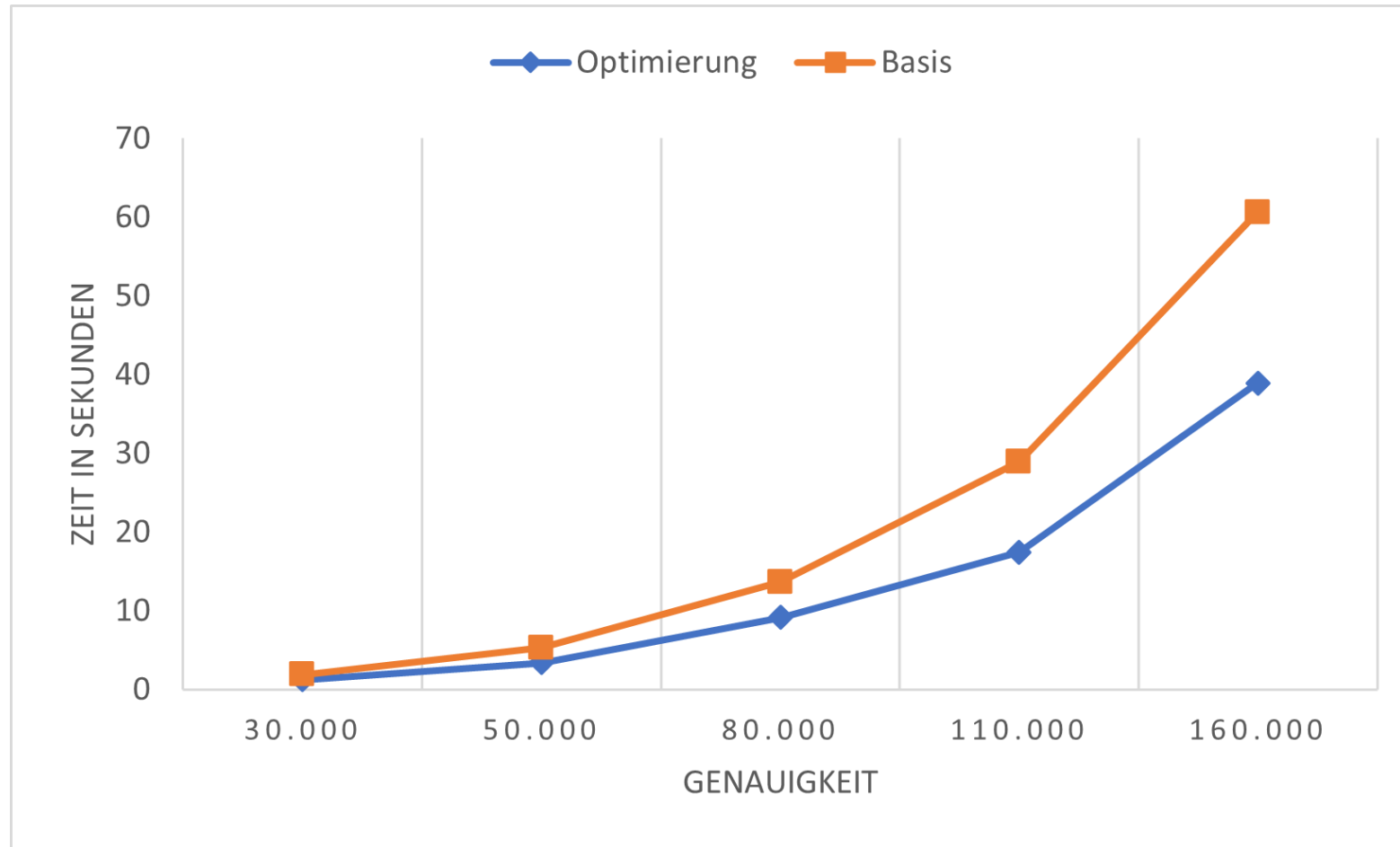
$$1 * 2^3 + 1 * 2^2 + 0 * 2^1 + 1 * 2^0$$

$$7^{2^3+2^2+2^0} = 7^{2^3} * 7^{2^2} * 7^{2^0}$$

# Performance

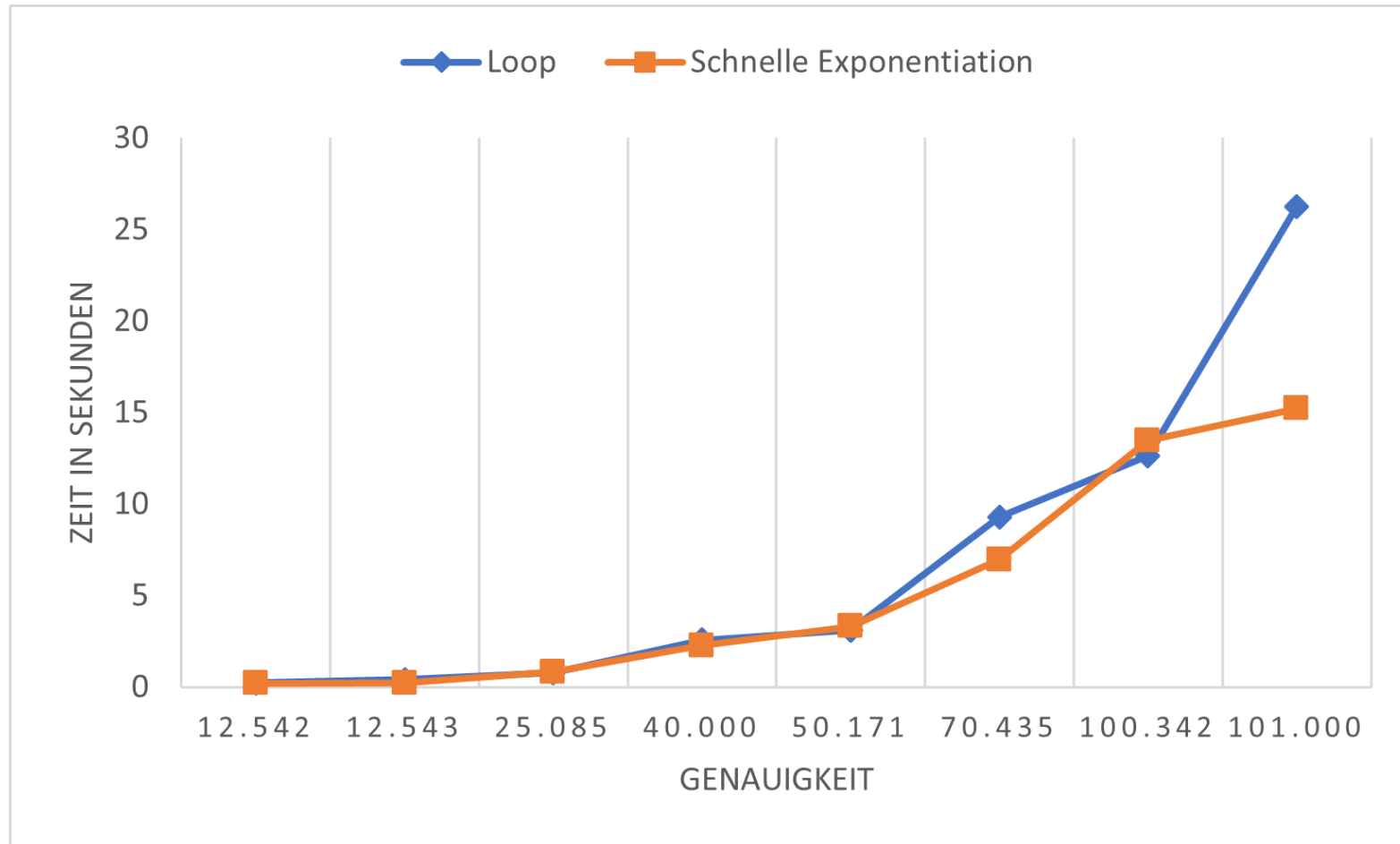


# Performance





# Performance



# Ausblick

- Limitierungen
- Größe der Zahlen
- Optimierungen
- Notwendigkeit

# Quellen

- <http://www.rankingshuttle.de/der-pagerank-algorithmus/>
- <https://cosmosmagazine.com/science/mathematics/working-with-numbers-the-square-root-of-2/>
- [https://www.moodle.tum.de/pluginfile.php/4290945/mod\\_page/content/9/2\\_schaltungen\\_prozessorkomponenten\\_adder\\_sub.pdf](https://www.moodle.tum.de/pluginfile.php/4290945/mod_page/content/9/2_schaltungen_prozessorkomponenten_adder_sub.pdf)
- [https://www.toppr.com/guides/computer-science/programming-in-c-/structured-data-type/introduction-to-arrays/#Introduction to Arrays](https://www.toppr.com/guides/computer-science/programming-in-c-/structured-data-type/introduction-to-arrays/#Introduction%20to%20Arrays)  
[programiz.com/c-programming/c-arrays](https://programiz.com/c-programming/c-arrays)