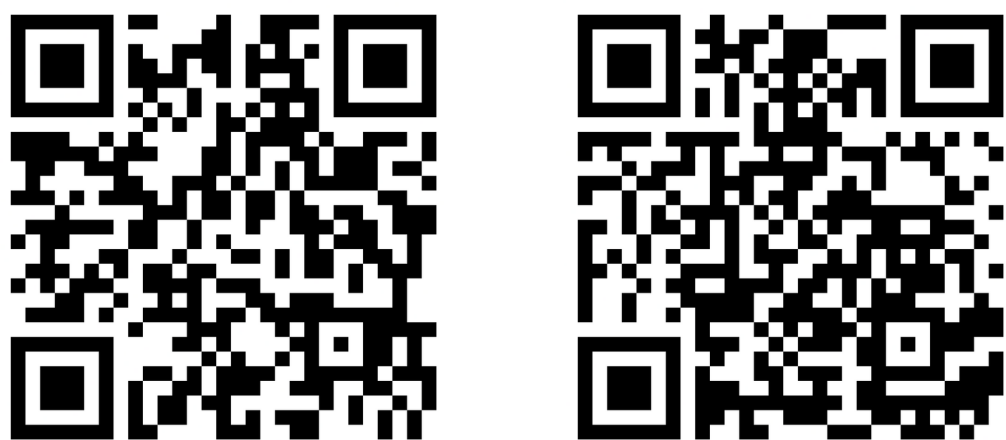


Wie beeinflusst die Wahl der Datenstruktur die Geschwindigkeit einer Datenbank und inwiefern eignet sich eine Datenstruktur für gewisse Daten besser oder schlechter?

Motivation

- Über den Informatik-LK hinaus mit Datenbanken und Daten-Modellierung beschäftigen
- Interne Funktionsweise von Datenbanken genauer kennenlernen
- Frage klären, welche Datenbanken/Datenstrukturen für welche Daten am besten sind
- Interesse durch Vortrag von Richard Hipp



Kurzfassung

In meinem Projekt wollte ich erforschen, wie die in Datenbanksystemen genutzten Datenstrukturen die Zugriffsgeschwindigkeit beeinflussen und welche Datenstrukturen sich für welche Arten von Daten besonders gut eignen. Dazu wollte ich ein Datenbanksystem in C++ entwickeln und messen, wie schnell, je nach Datenstruktur, auf die Daten zugegriffen werden kann. Die Haupt-Schwierigkeit war dabei allerdings die hohe Komplexität, weshalb ich nur einige Datenstrukturen umsetzen und testen konnte. Insgesamt lässt sich feststellen, dass noch weitere Messungen nötig sind, um die Ergebnisse zu konsolidieren. Dies lässt noch viele weitere Möglichkeiten für weitere Forschungen zu.

Binäres-Sucharray

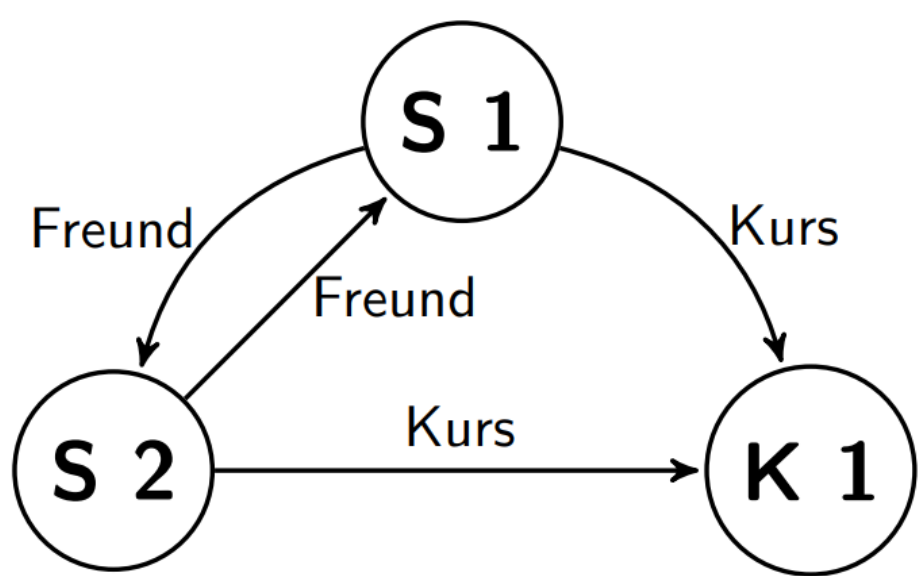
- gleicher Suchalgorithmus wie B-Tree/Binary-Tree
- Testdaten auf 3 Such-Arrays (vgl. Tabellen) verteilt
- für eine Bereichssuche muss 2x die binäre Suche angewand werden

Test mit Testdaten

- Datenstrukturen im RAM und nicht auf einem Massenspeicher (e.g. SSD)
- **400.000** Schüler, jeweils **100** Freundschaften/Kursbelegungen
- Getestet wurde die Zeit zur Suche aller Freunde eines Schüler (tiefgestellt **F**) und die Zeit zur Suche aller Kursbelegungen (tiefgestellt **H**). Diese Operationen sind dabei gleichwertig und sollten gleich lange dauern
- Ich vermutete, dass die Graph-Datenstruktur schneller ist, weil dort der Suchalgorithmus deutlich einfacher ist
- SA bezeichnet den Graph des Sucharrays, G von der Graph-Datenstruktur

Arten von Daten und Datenstrukturen

- Räumliche Daten sind z. B. Kartendaten (Google Maps, OpenStreetMap), **R-Tree**
- Zeitliche Daten können z. B. die von einem Sensor gemessenen Temperaturen sein
- Objekte z. B. Schüler, häufig in relationalen Datenbanken gespeichert, **B-Tree/Binary-Tree**
- Beziehungs- und Graphdaten z. B. auf Social-Media Plattformen, **Graph-Datenstruktur**



Schüler_1_ID	Schüler_2_ID
5	7
5	75
5	456
5	45

Schüler_1_ID	Kurs_ID
7	629
7	762
7	229
7	29435

Arten der Modellierung und Datenspeicherung

Graph-Datenstruktur

- In jedem Node sind die Verbindungen(Freundschaften/Kursbelegungen) eines Schülers gespeichert
- Iteration über die Elemente zur Suche

$$f_g = \text{Anzahl Freundschaften} \quad K_g = \text{Kursbelegungen}$$

$$f = \text{Anzahl Freunde eines Schülers} \quad K = \text{Anzahl Kurse eines Schülers}$$

$$s = \text{Anzahl Schüler}$$

$$O(2 \cdot (\log_2(f_g) + 1) + f) \quad O(f + K)$$

Komplexitätsbetrachtungen für Binäres-Sucharray (l.) und Graph-Datenstruktur (r.)

Testergebnisse von Binärem-Sucharray und Graph-Datenstruktur

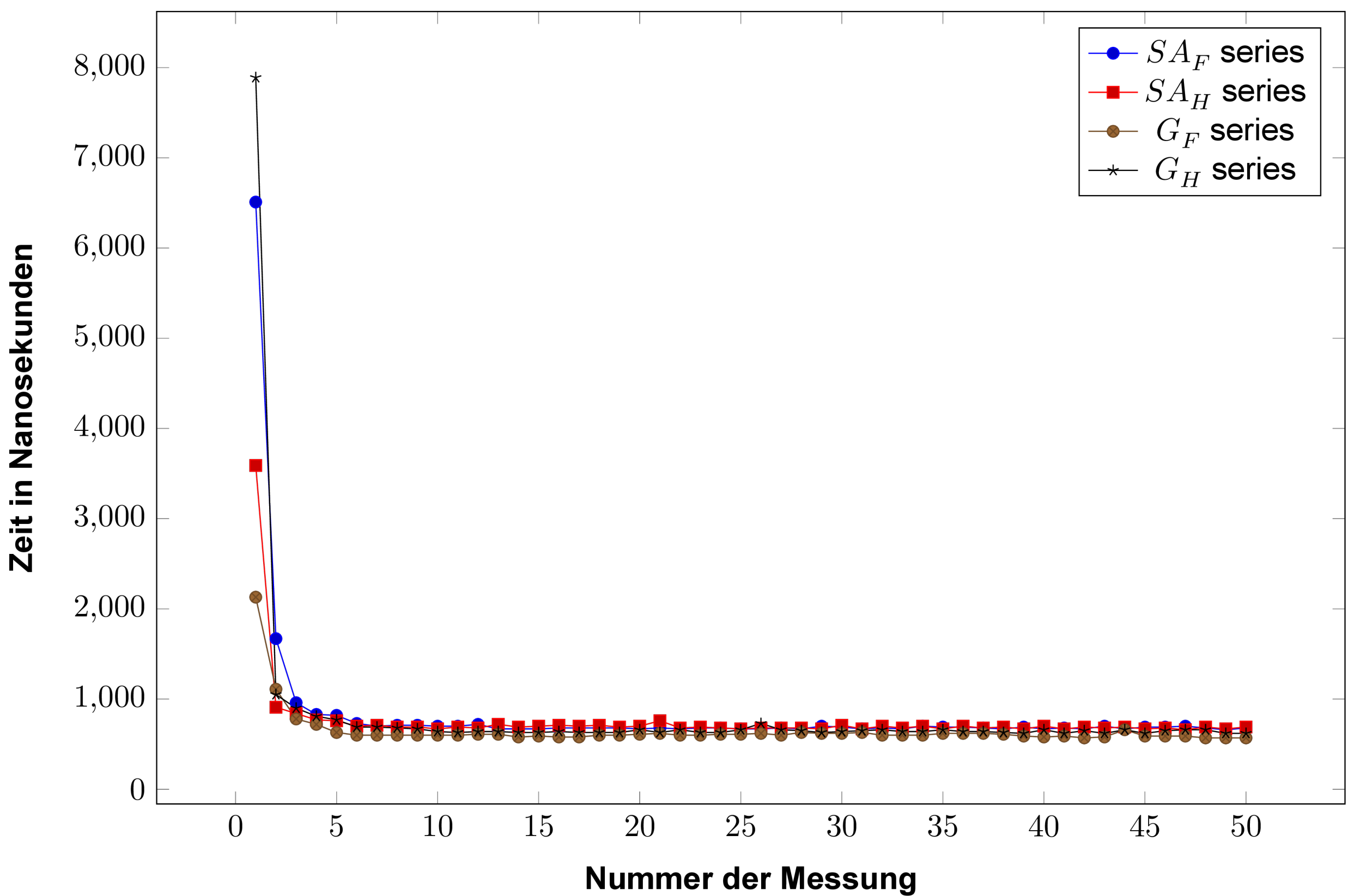


Abbildung 5: 1. Messreihe am PC (Ryzen 7 2700)

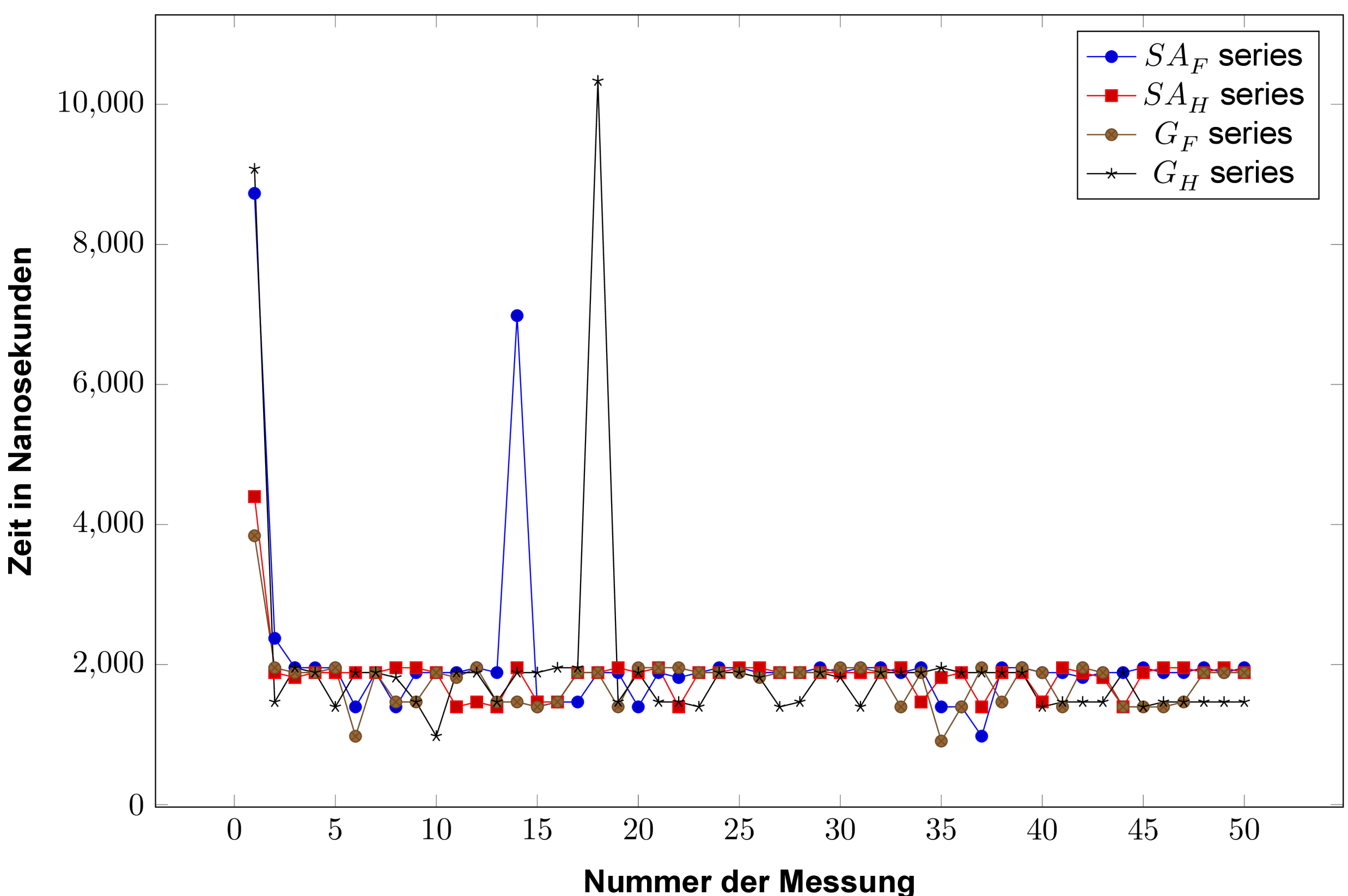


Abbildung 7: 1. Messreihe am Laptop (Ryzen 5 5500U)

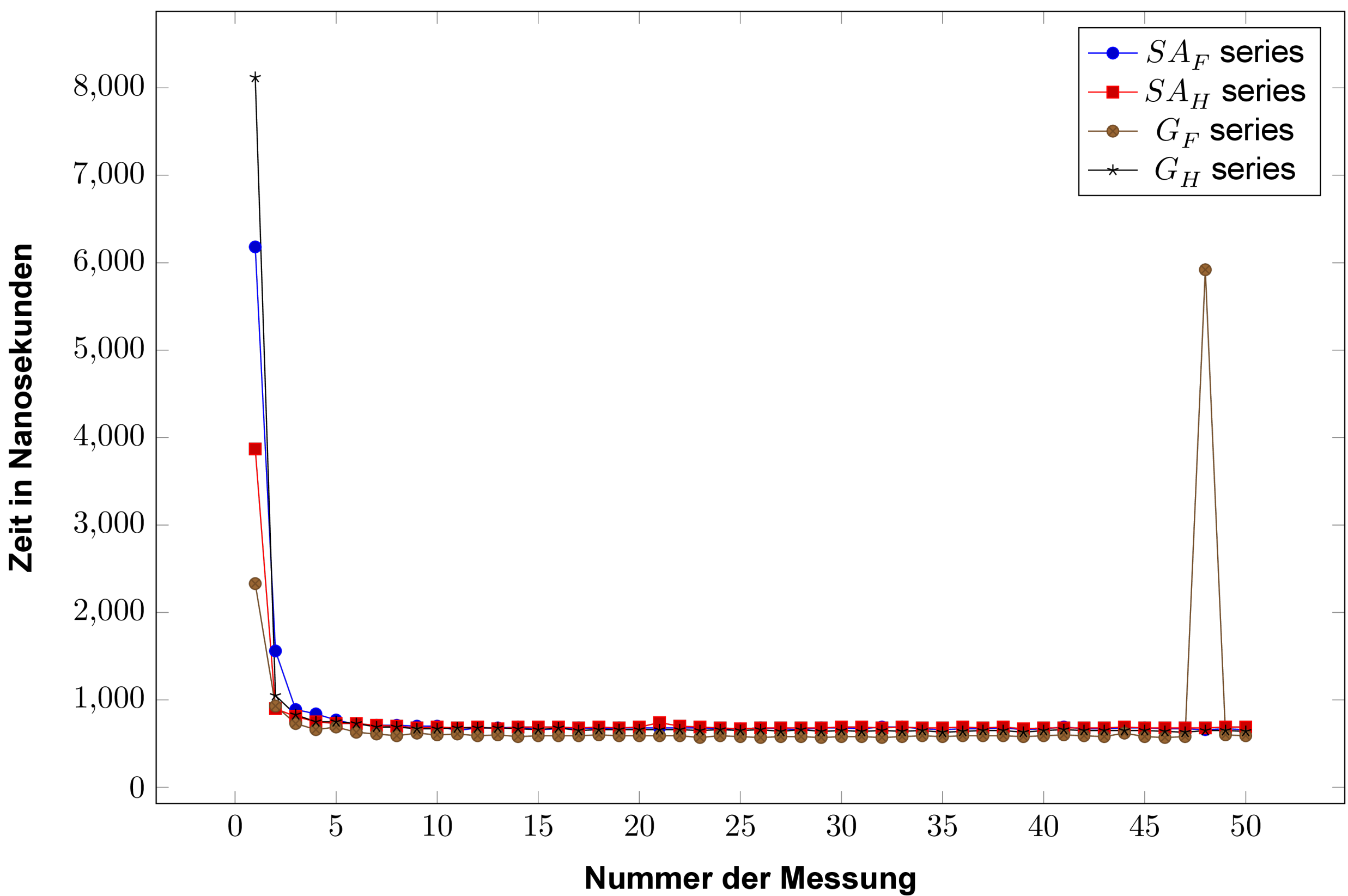


Abbildung 6: 2. Messreihe am PC (Ryzen 7 2700)

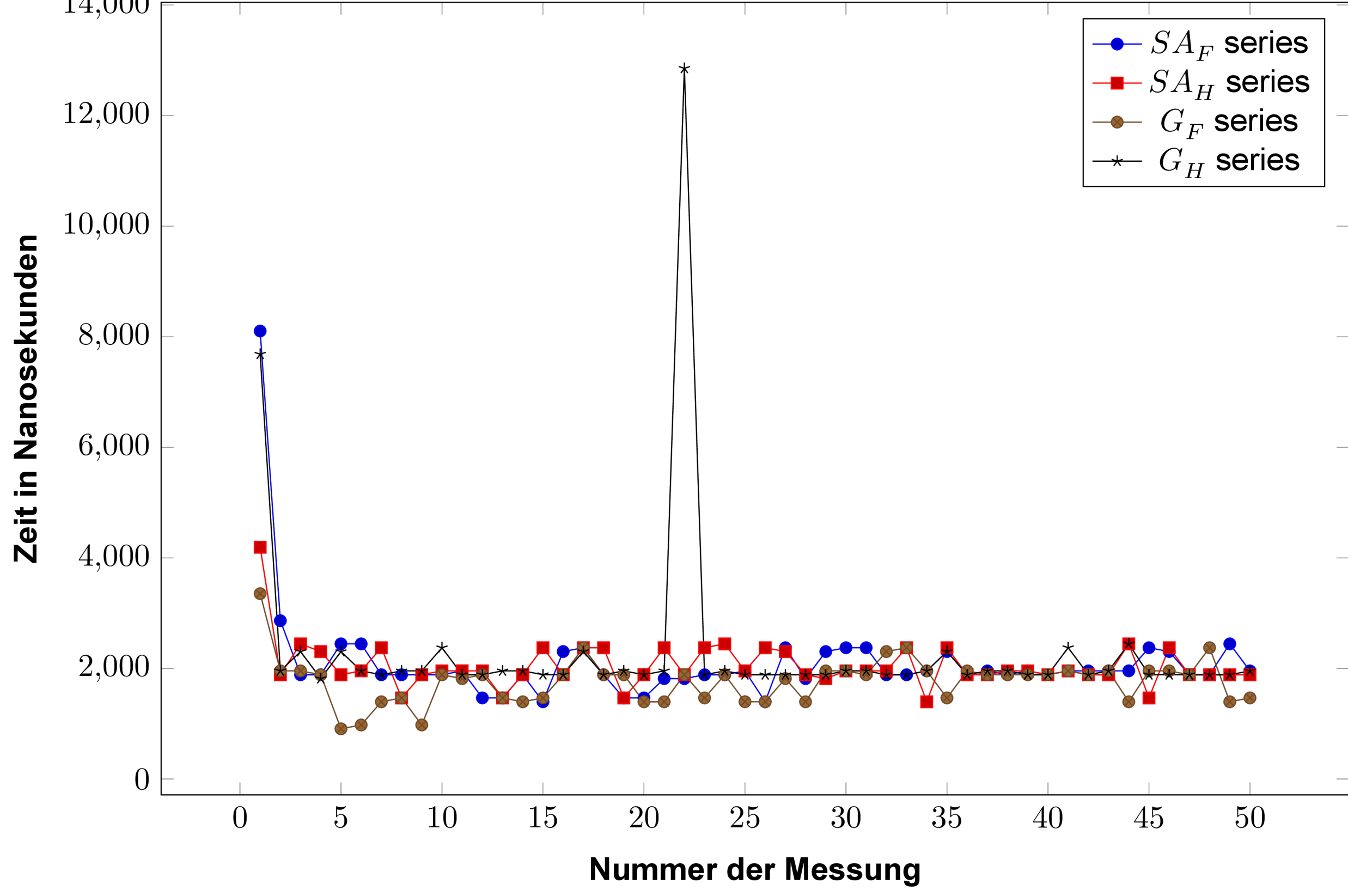


Abbildung 8: 2. Messreihe am Laptop (Ryzen 5 5500U)