



Hochschule für Angewandte Wissenschaften Hamburg

Hamburg University of Applied Sciences

Algorithmen & Datenstrukturen

Aufgabenblatt 2

Alexander Mendel
Karl-Fabian Witte
Florian Heuer

Inhaltsverzeichnis

1. Einleitung.....	3
2. Empirische Untersuchung.....	3
2.1 Langsames Suchverfahren.....	3
2.2 Schnelles Suchverfahren.....	4
2.3 Sieb des Eratosthenes.....	5
2.4 Einzelnes Suchverfahren.....	6
3. Fazit.....	7
4. Repository.....	7

1. Einleitung

Das folgende Dokument soll die empirische Untersuchung der in der Aufgabenstellung beschriebenen Algorithmen zum Finden von Primzahlen dokumentieren.

In Punkt 2 werden alle Ergebnisse tabellarisch sowohl als auch grafisch dargestellt und dazu beziehend die Komplexität der Algorithmen bestimmt.

2. Empirische Untersuchung

Die untersuchten Problemgrößen sind 10^K . Für $K = 1, 2, 3, 4$ und 5 . Die Problemgröße N wird in den folgenden Diagrammen auf der X-Achse abgetragen. Der Aufwand $T(N)$ der Algorithmen auf der Y-Achse.

2.1 Langsames Suchverfahren

In der nebenstehenden Tabelle 1 sind die für die Problemgrößen ermittelten Aufwände gelistet.

Was an den Werten noch nicht unbedingt klar deutlich wird, ist in der Abb. 1 zu sehen. Der Aufwand wächst mit steigender Größe N quadratisch an.

N	T(N)
$10^K \mid K=1$	72
$10^K \mid K=2$	9.774
$10^K \mid K=3$	1.006.776
$10^K \mid K=4$	100.976.778
$10^K \mid K=5$	10.100.676.780

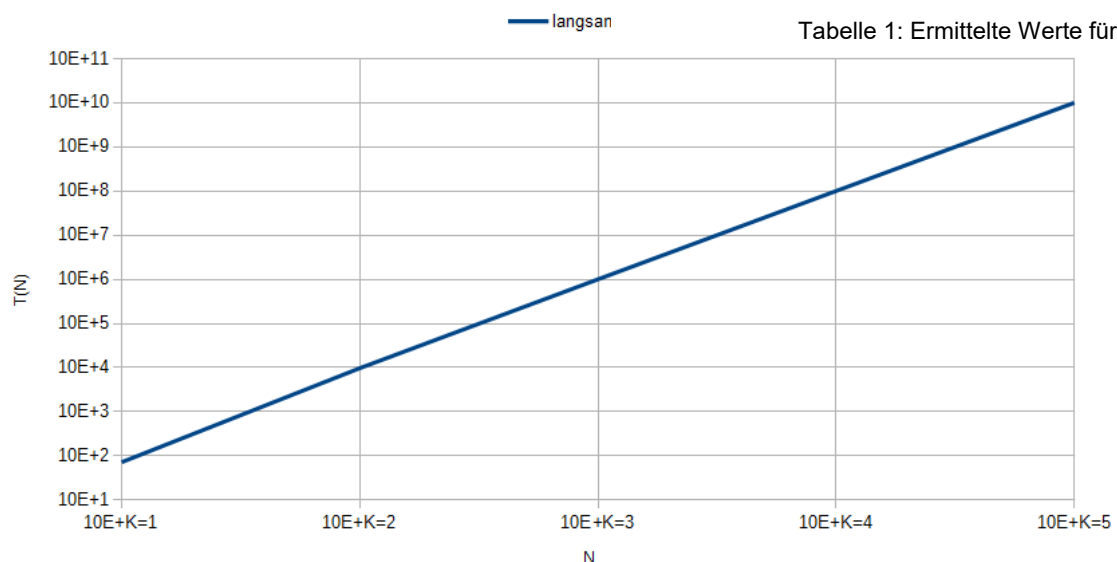


Abbildung 1: Langsames Suchverfahren mit der Komplexität $O(n^2)$

2.2 Schnelles Suchverfahren

In der nebenstehenden Tabelle 2 sind die für die Problemgrößen ermittelten Aufwände gelistet.

Was an den Werten auffällt wenn man diese mit den Werten aus Tab. 1 vergleicht, ist dass, die aus Tab. 2 nur halb so groß sind. D.h. eine Verbesserung des Algorithmus wurde erzielt.

Schaut man jedoch auf die Abb. 2 so ist kaum eine Verbesserung ersichtlich, denn die Komplexität bleibt trotz Verbesserung ebenfalls bei $O(n^2)$.

N	T(N)
$10^K \mid K=1$	36
$10^K \mid K=2$	4.887
$10^K \mid K=3$	503.388
$10^K \mid K=4$	50.488.389
$10^K \mid K=5$	5.050.338.390

Tabelle 2: Ermittelte Werte für N

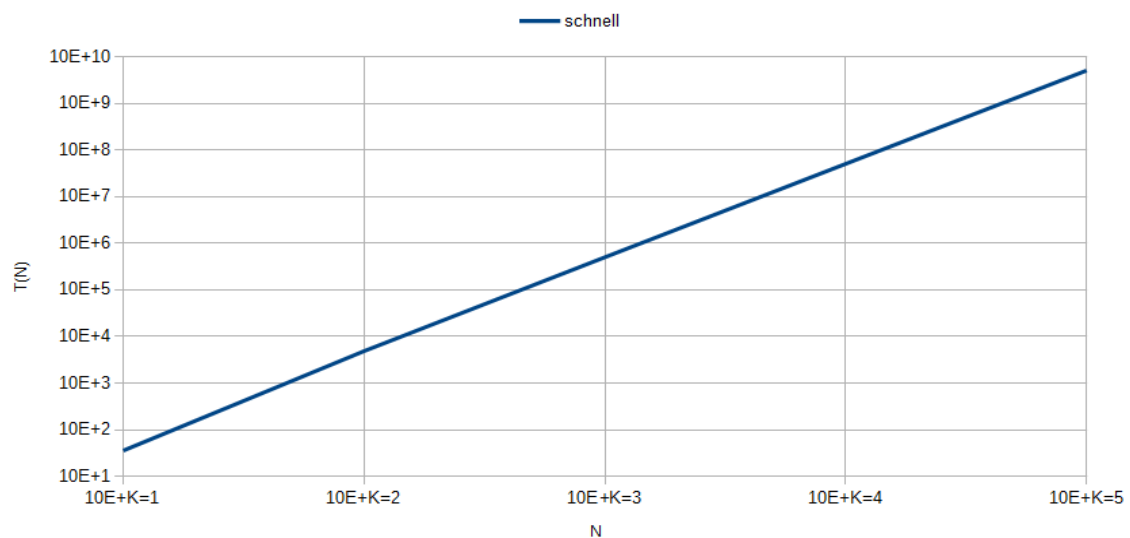


Abbildung 2: Schnelles Suchverfahren mit der Komplexität $O(n^2)$

2.3 Sieb des Eratosthenes

In der nebenstehenden Tabelle 3 sind die für die Problemgrößen ermittelten Aufwände gelistet.

Zu erkennen ist an den Werten, dass der Algorithmus deutlich weniger Aufwände erzeugt als die zuvor vor gezeigten. Der Graph verläuft deutlich flacher und ist weit von den Aufwänden aus Abb.1 & 2 entfernt.

N	T(N)
$10^K \mid K=1$	4
$10^K \mid K=2$	123
$10^K \mid K=3$	1.668
$10^K \mid K=4$	19.755
$10^K \mid K=5$	222.221

Tabelle 3: Ermittelte Werte für N

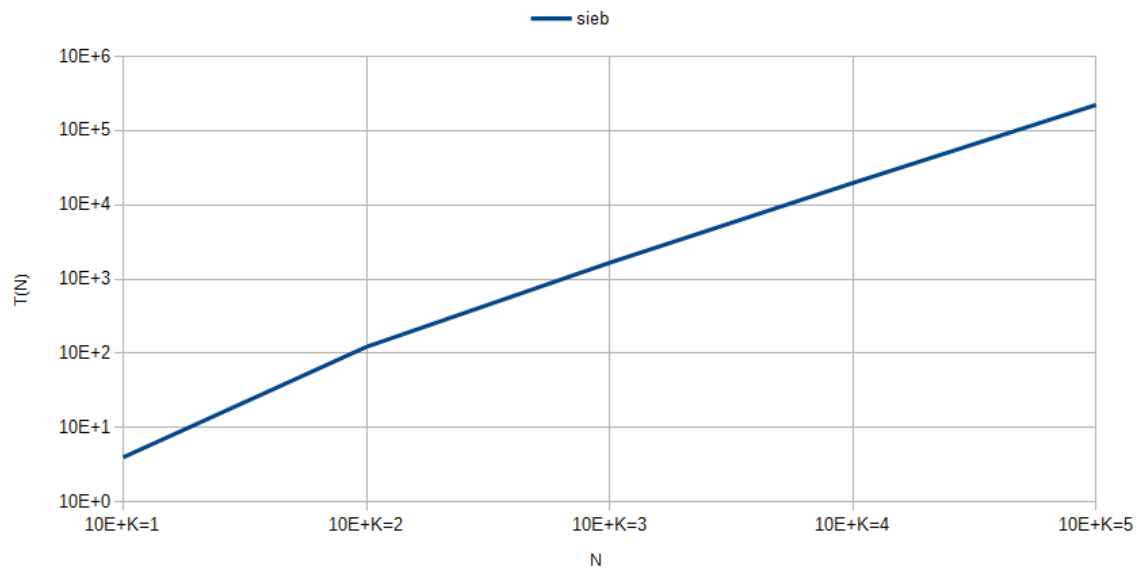


Abbildung 3: Sieb des Eratosthenes mit der Komplexität $O(n \log(n))$

2.4 Einzelnes Suchverfahren

In der nebenstehenden Tabelle 4 sind die für die Problemgrößen ermittelten Aufwände gelistet.

Vermeintlich ist an den Werten sehen, dass dieser Algorithmus deutlich besser abschneidet als 2.1 und 2.2. Das liegt daran dass, die Aufwände aus der aufrufenden Schleife nicht mit gezählt werden, weil sie kein Teil des Algorithmus' zum finden einer Primzahl sind.

An der Steigung des Graphen ist aber trotzdem erkennbar dass, die Komplexität bei $O(n^2)$ liegen muss.

N	T(N)
$10^K \mid K=1$	14
$10^K \mid K=2$	1.146
$10^K \mid K=3$	79.167
$10^K \mid K=4$	5.854.389
$10^K \mid K=5$	461.043.538

Tabelle 4: Ermittelte Werte für N

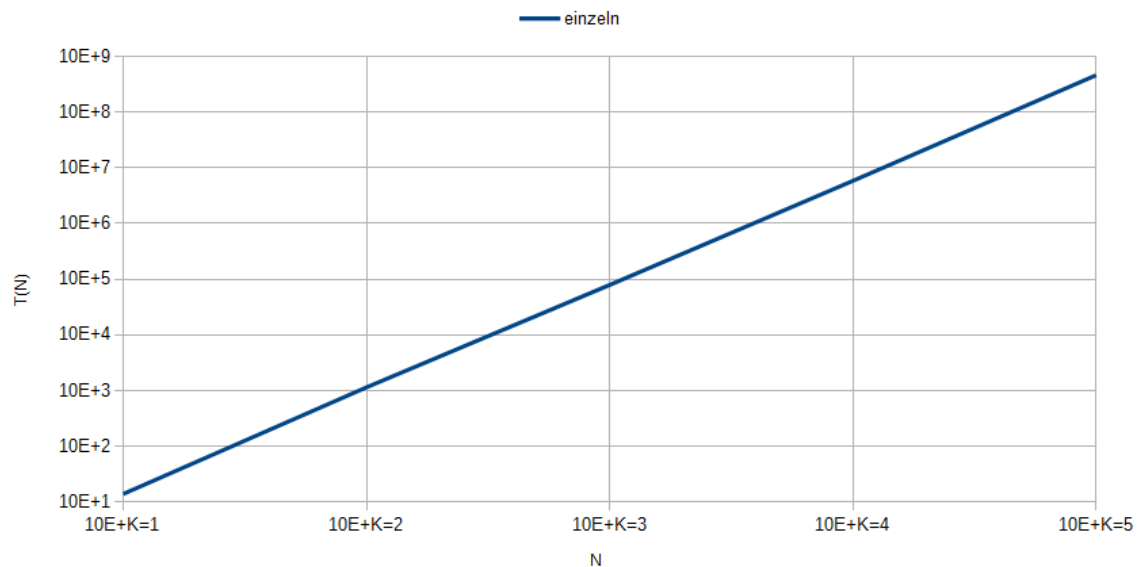


Abbildung 4: Einzelne Primzahlssuche mit der Komplexität $O(n^2)$

3. Fazit

Zum Abschluss sieht man nun in der folgenden Abbildung 5 alle zuvor untersuchten Algorithmen vereint in einem Diagramm. Aus diesem wird noch deutlicher ersichtlich, dass der Algorithmus aus 2.3 deutlich besser abschneidet als die Restlichen.

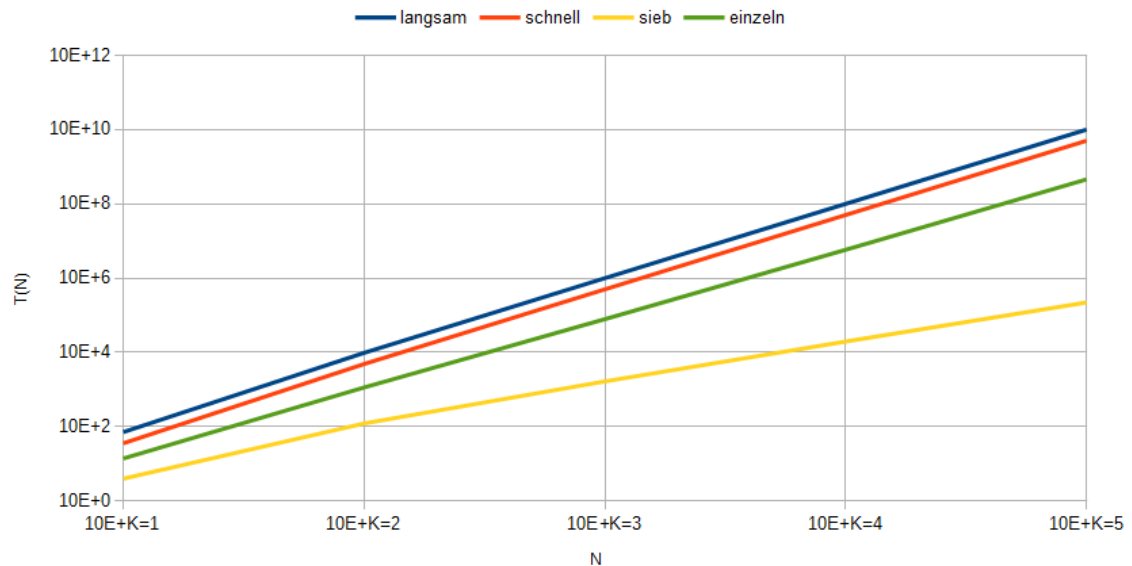


Abbildung 5: Vergleich aller Algorithmen

4. Repository

Der Quellcode zur Aufgabe ist im folgenden Repository zu finden.

<https://github.com/FlowwX/AD>