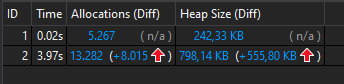
**ALGOS Plan**

Vektoren wurden für die Einträge der historischen Daten verwendet, da das Freien des Speichers dann übernommen wird.

Die Größe der Hashtabelle wurde so gewählt, dass die Tabelle unter 50% Auslastung bleibt, da sonst die quadratische Sondierung eventuell keinen freien Platz mehr finden kann. Falls die Größe keine Primzahl ist passiert kann dies sogar schon unter 50% passieren.

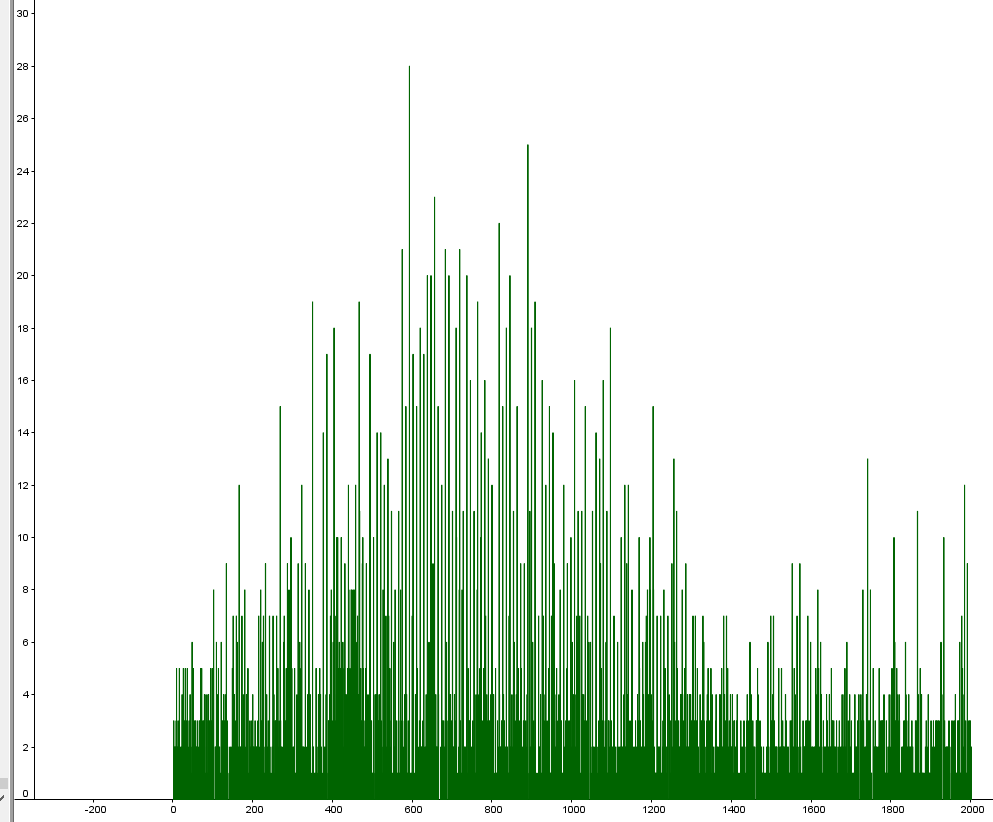
Die Hashtabelle wird nicht mit new allokiert, da sie sonst auf dem Heap gespeichert wird. Der Stack ist schneller. EDIT: die Hashtabelle wird mit new allkoiert, da sich sonst VS aufregt, dass zu viel am Stack gespeichert wird (~500 KB).

Bezüglich der Allokierung der einzelnen stocks: Man könnte sie auch erst allokieren, wenn man die einzelnen index benötigt, jedoch würde bei der löschung die node bestehen bleiben müssen, da sonst ihr deleted attribute nicht gesetzt werden kann, was im endeffekt einfach nur zu random allokierten nodes führen würde. Deswegen allokieren wir sie quasi on startup. Ist zwar eine größere speicherauslastung, aber vereinfacht die handhabung

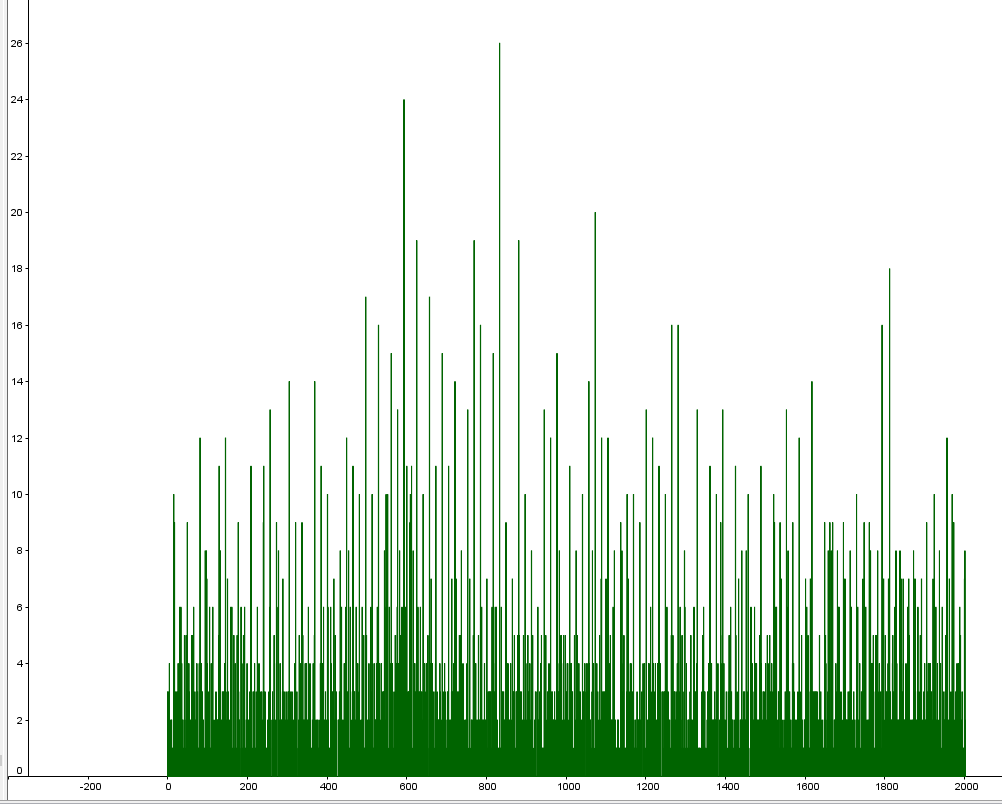


Evaluierung der Hashfunktion:

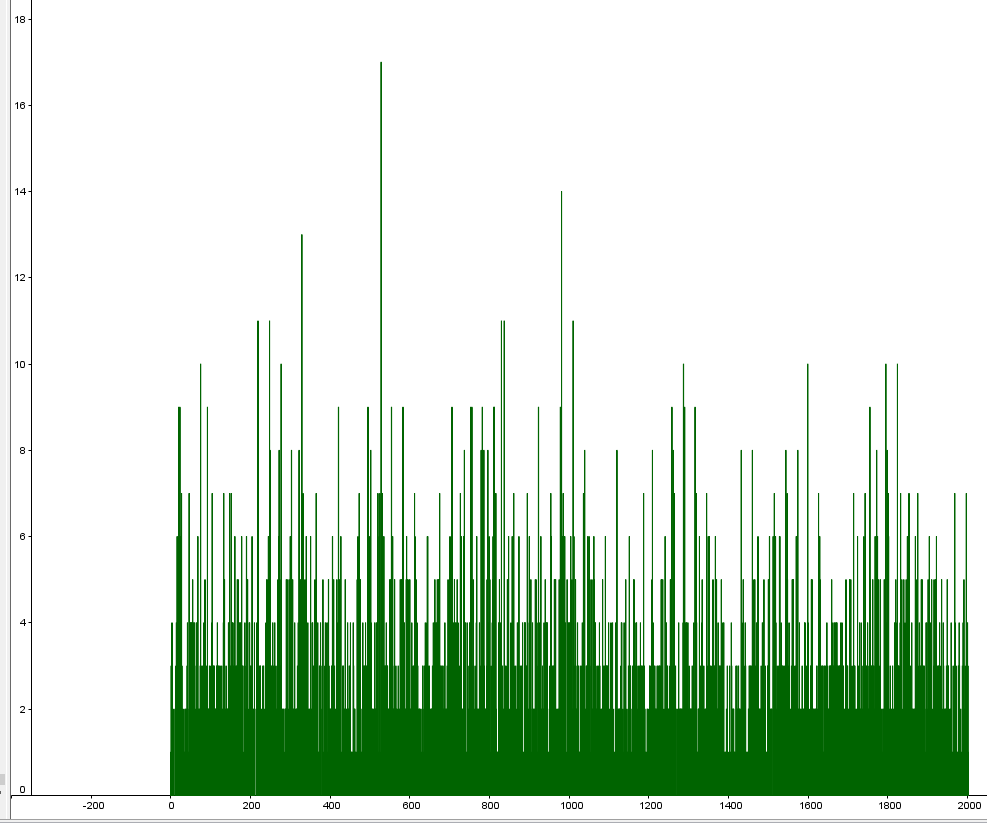
Mit 3er Potenz:



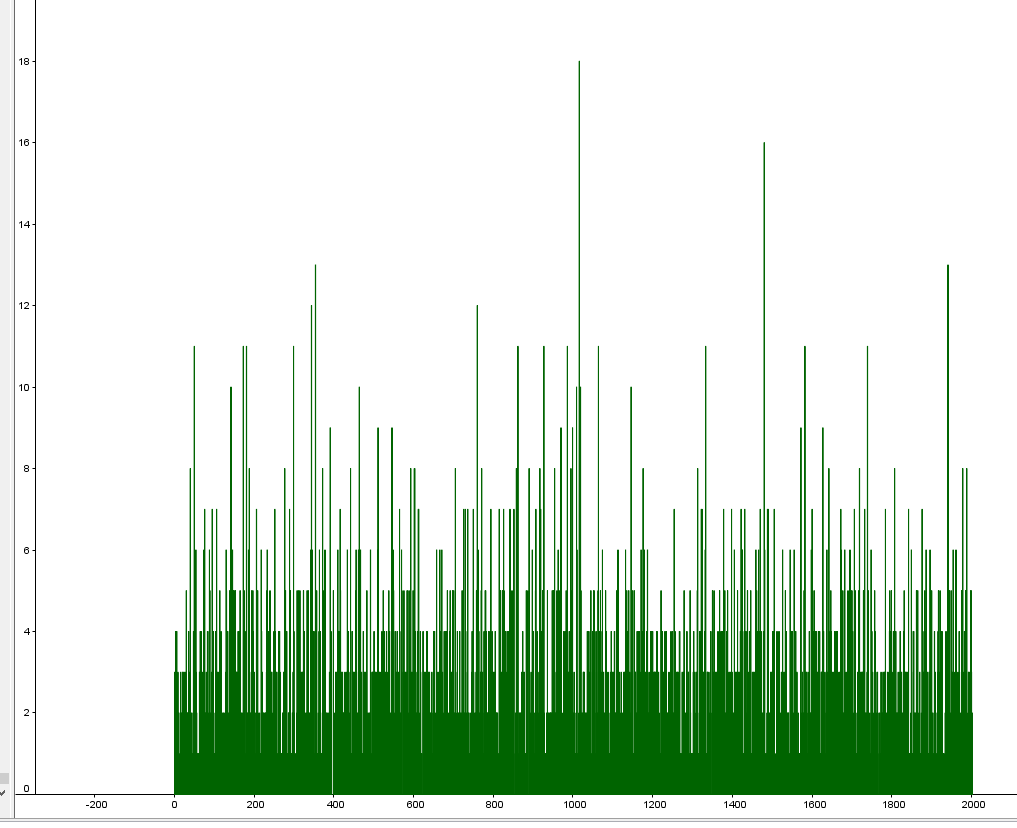
Mit 4er Potenz:



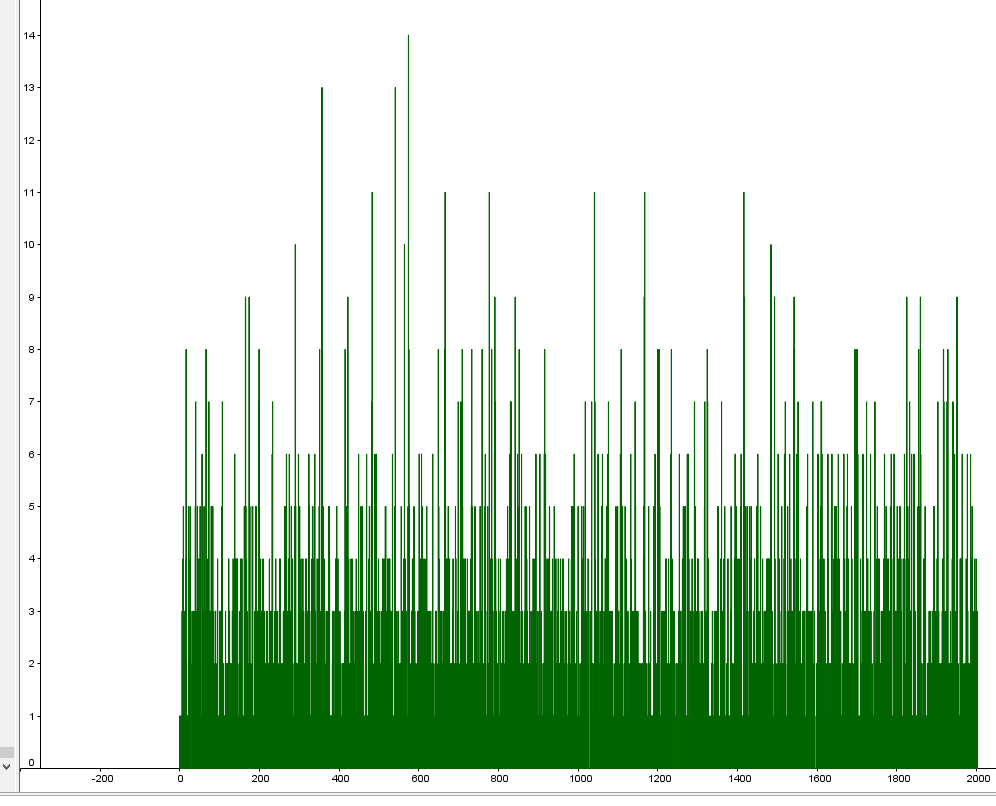
Mit 31er Potenz:



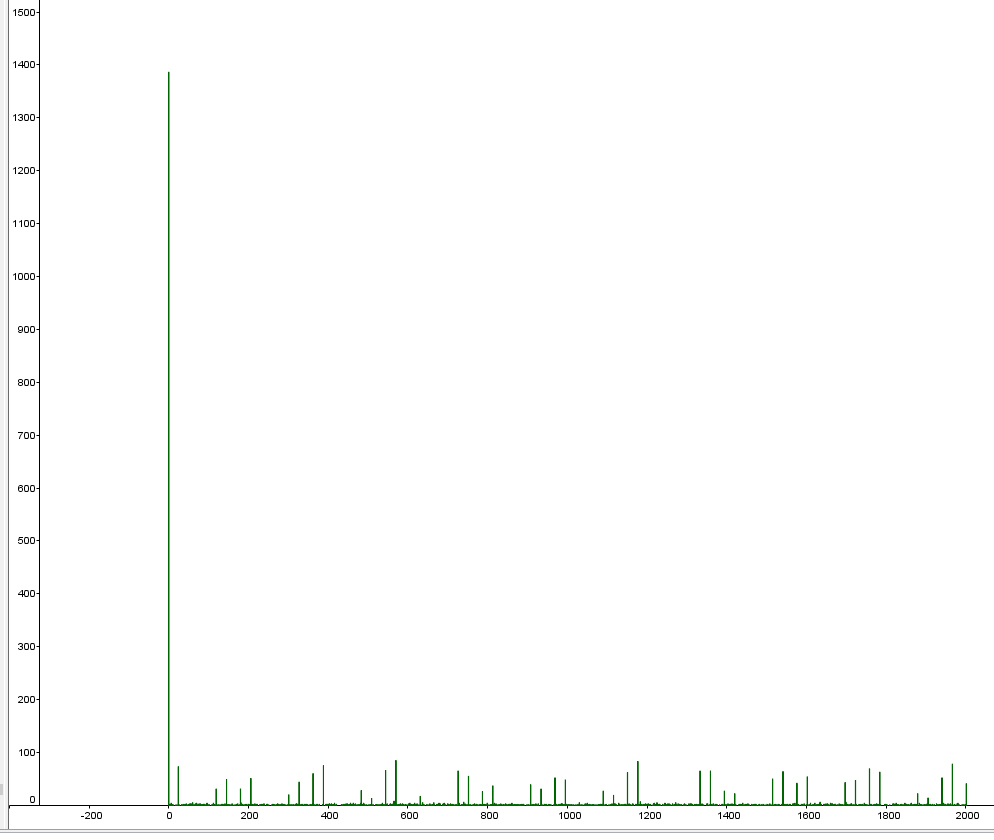
Mit 32er Potenz:



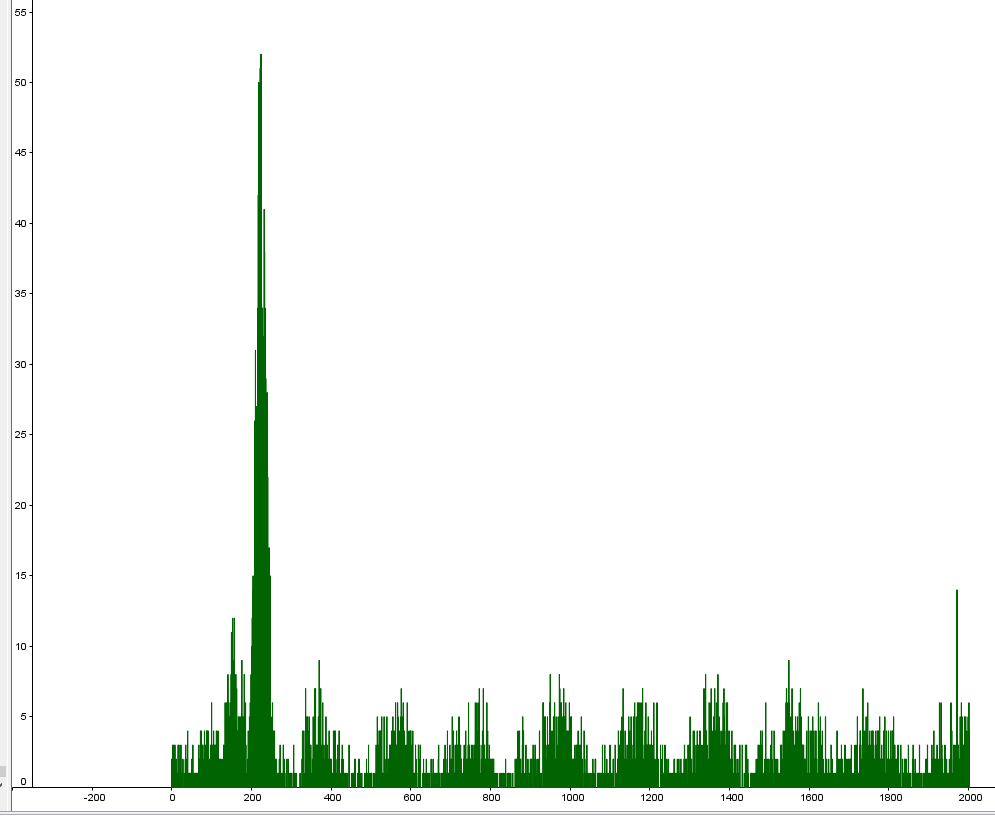
Mit 71er Potenz:



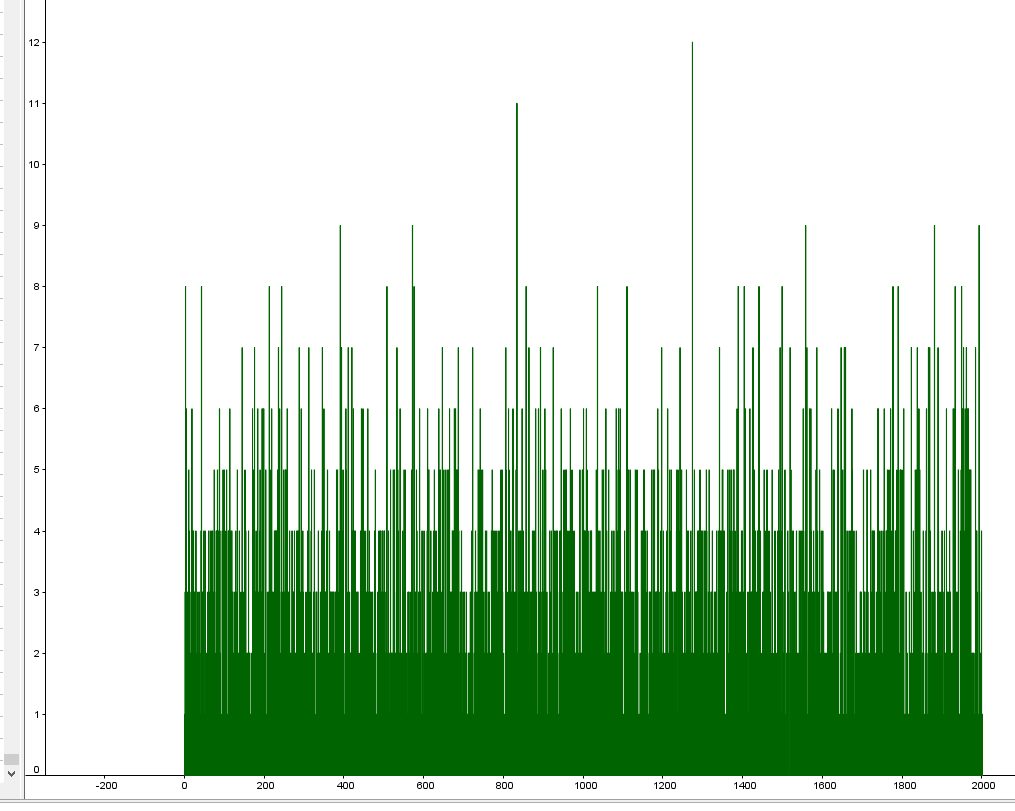
Mit 2003er Potenz:



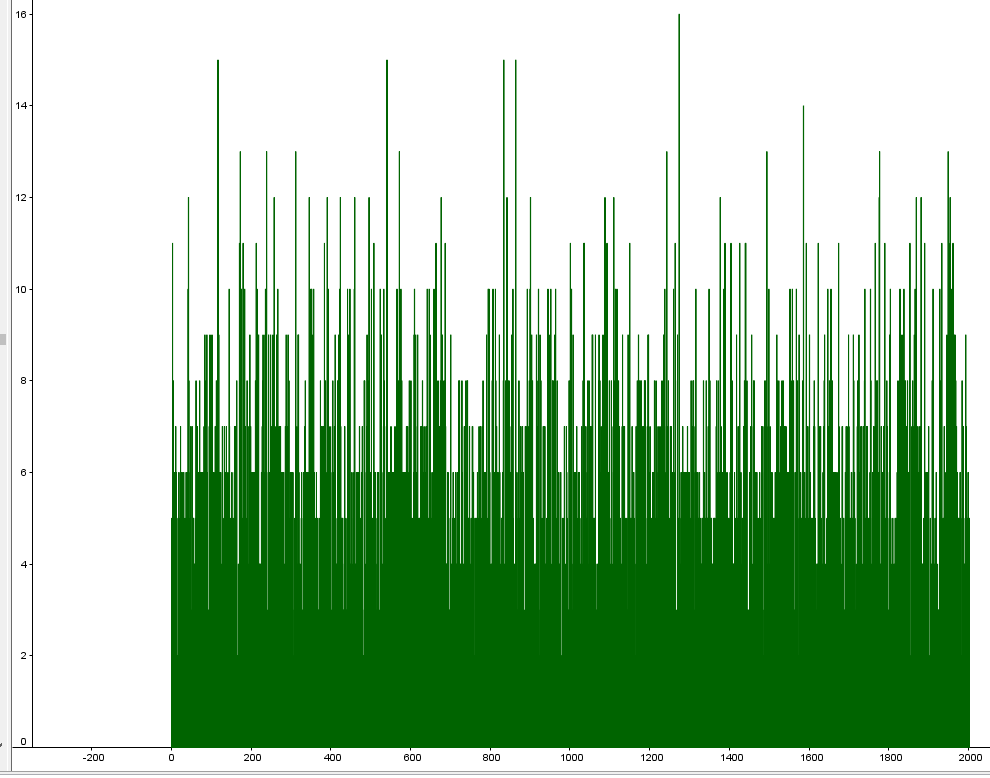
Mit 2004er Potenz:



Mit mutiplizieren von jedem char:



Mit 9.999 Werten



Durchschnitt über 100.000: 