

External Mergesort

Finn Stutzenstein, Levin Nemesch, Joshua Sangmeister

16. November 2020

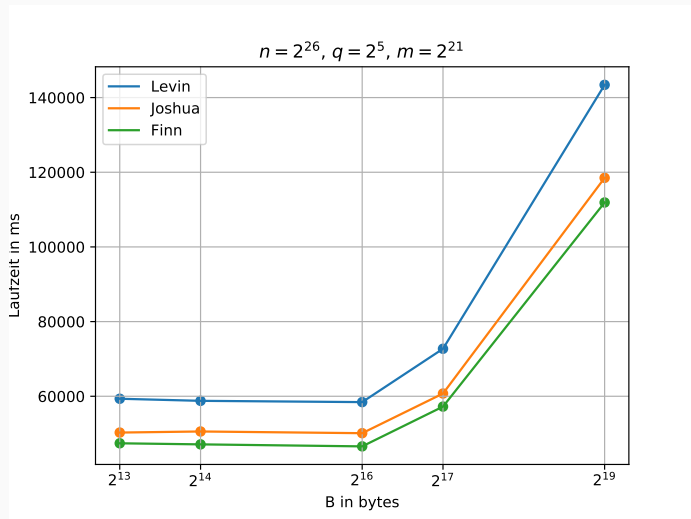
Algorithm Engineering - Übung 1

- Parameter
 - M: Größe des internen Speichers in Byte
 - B: Blockgröße in Byte
 - Q: Elementgröße in Bit
- Achtung: Effektive Anzahl der Elemente pro Block und im Speicher ist unterschiedlich bei variierendem Q.
- L1, L2, L3 Caches:

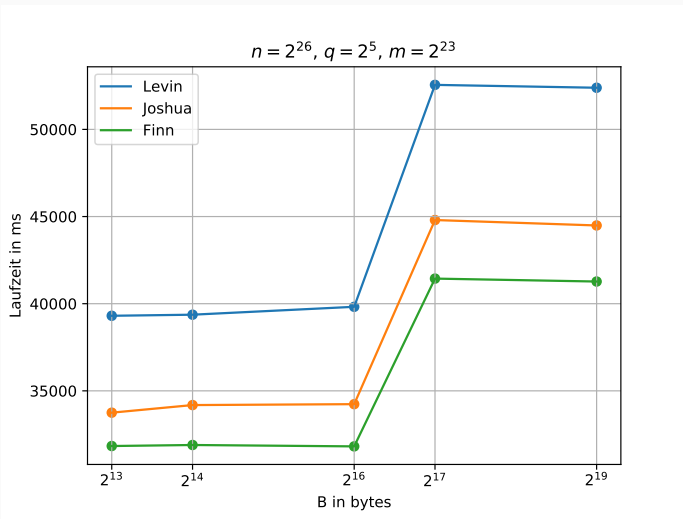
	CPU	L1	L2	L3
Finn	i5-3360M	64 KiB	512 KiB	3 MiB
Levin	i5-5200U	64 KiB	512 KiB	3 MiB
Joshua	i5-8265U	128 KiB	1 MiB	6 MiB

- Code: <https://github.com/jsangmeister/AE/tree/master/ue/ue1>

Sehr große Blöcke (128k und 512k) erhöhen Laufzeit.



Sehr große Blöcke (128k und 512k) erhöhen Laufzeit.

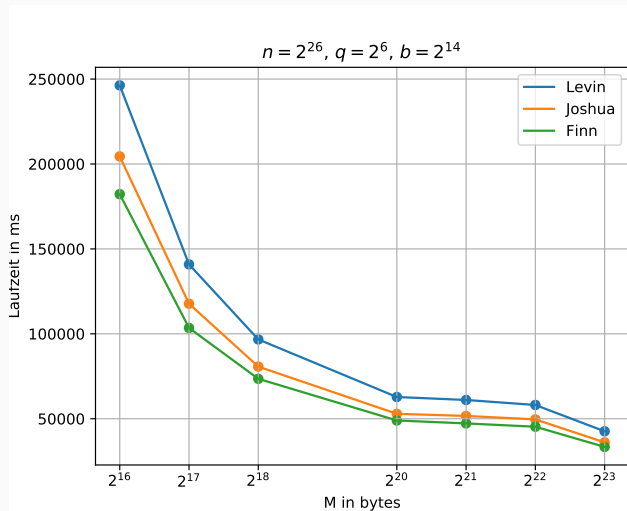


Sehr große Blöcke (128k und 512k) erhöhen Laufzeit. Warum?

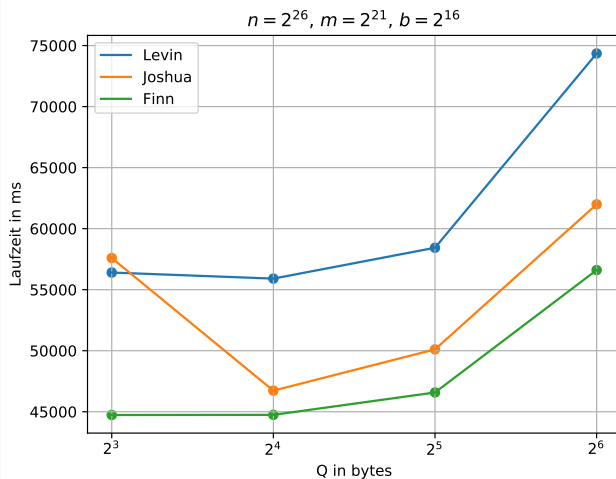
- Tatsächliches B der Systeme wird überschritten
- Größe der schnellen Caches (L1) wird überschritten

Memory

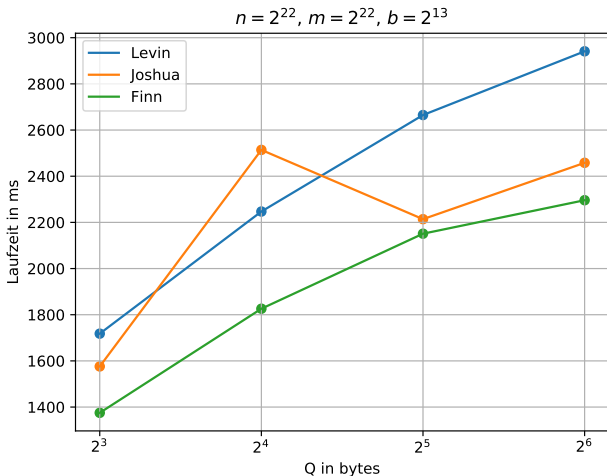
Bei mehr internem Speicher wie erwartet schneller



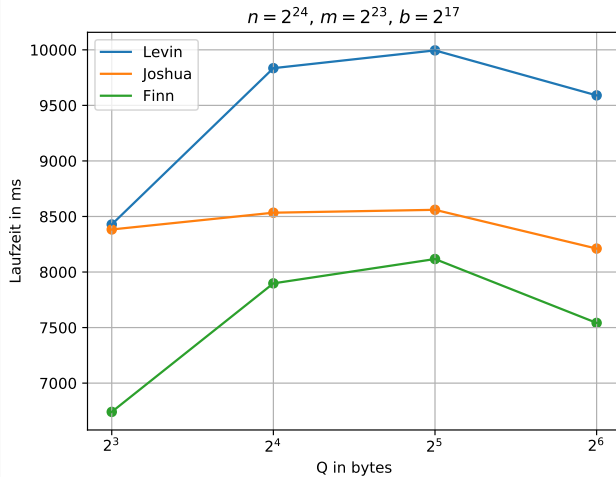
Größere Elemente bedeutet für großes n weniger Elemente pro Block und im Speicher, d.h. die effektive Block- und Speichergröße sinkt.



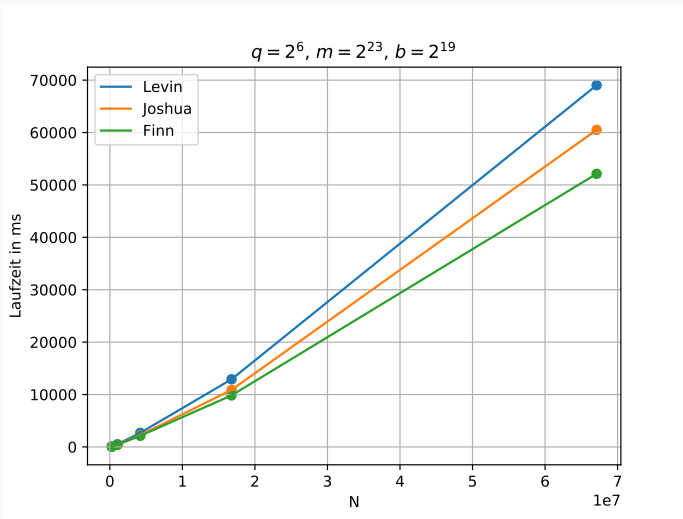
Joshua hat bei $n = 2^{22}$, $Q = 8$ und $Q = 16$ durchweg erhöhte Laufzeiten. Die Ursache ist nicht klar.



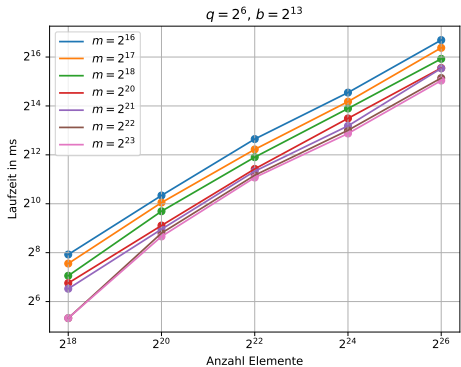
Ein ähnlicher Effekt zeigt sich bei $n = 2^{23}$ für uns alle



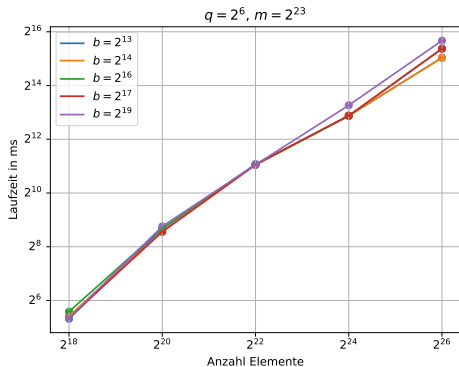
Wachsende Anzahl Elemente führt zu linearem Wachstum



(Daten von Finn)

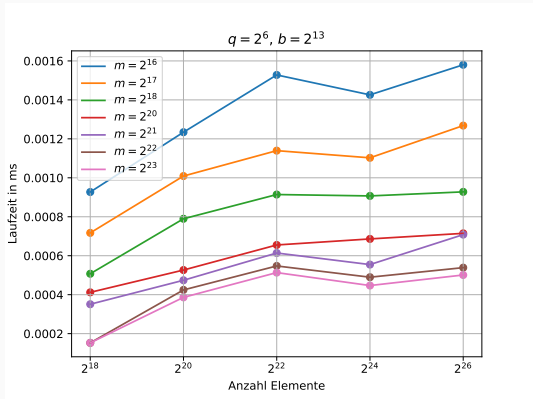


Bei größerem M ist Laufzeit niedriger,
unabhängig von N und B

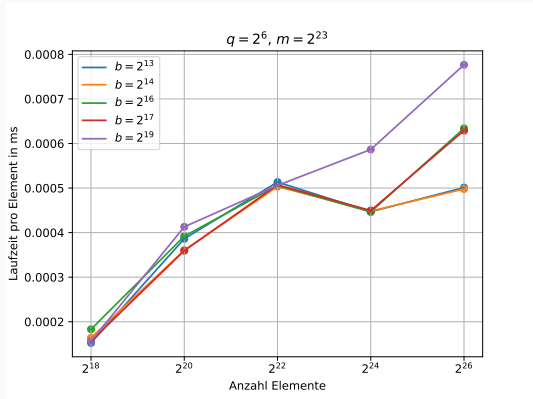


Bei fixem M hat B quasi keinen Einfluss auf die
Laufzeit

(Daten von Finn)



Laufzeit pro Element steigt minimal mit größerem N, aber nicht nennenswert
Größeres M führt auch zu geringerer Laufzeit pro Element



Laufzeit pro Element steigt mit größerem N bei fixem M
B ist quasi unerheblich, nur $B = 2^{19}$ ist 'zu groß'

- Für wachsendes n steigt die Laufzeit fast linear, der starke Logarithmus in der I/O-Komplexität zeigt sich