

SPP

Gruppe: Leonard Bongard, Duy Quang Nguyen, Egemen Ulutürk

Aufgabe 2)

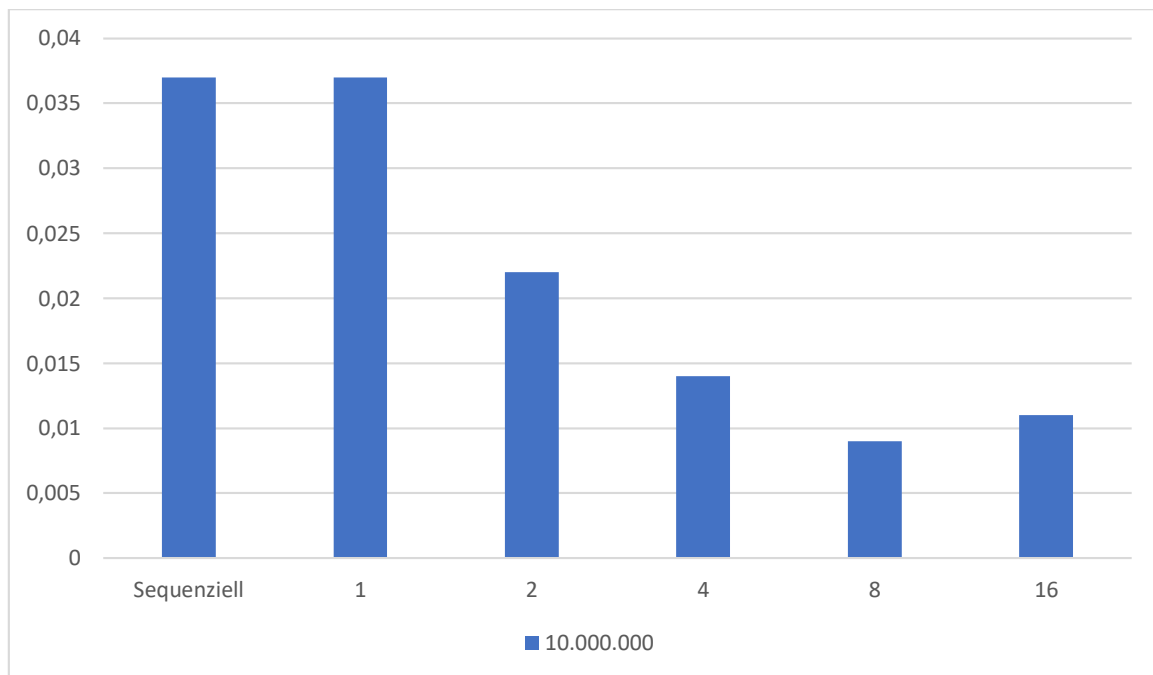
a)

1. i : private
2. j : shared
3. g1 : private
4. g2 : shared

b)

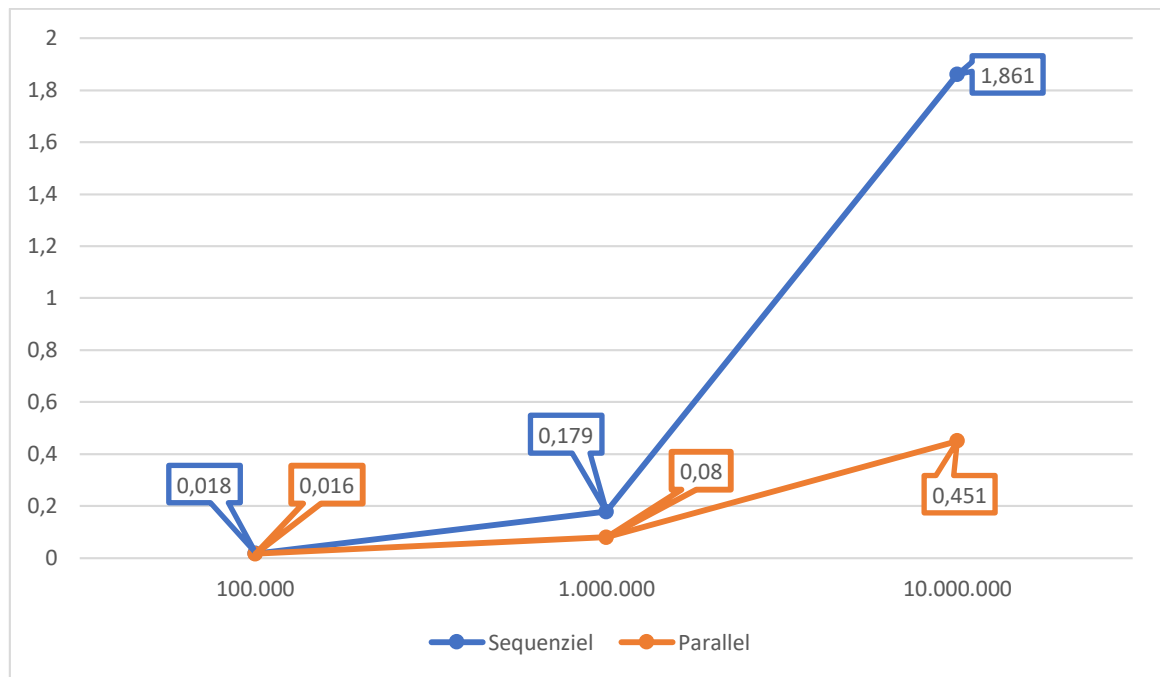
- P : private
g1 : private
g2 : shared

Aufgabe 3)



Aufgabe 4)

d)



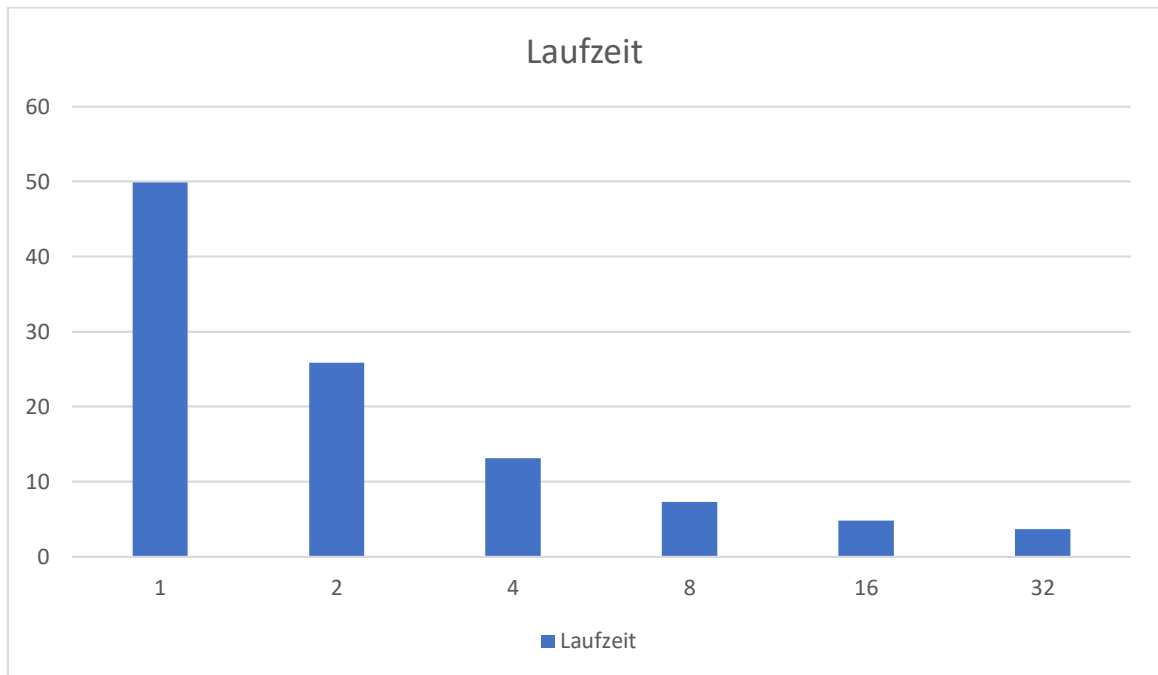
e)

Damit man „rand()“ in parallelisierten Code-Blöcken benutzen kann, muss diese eintrittsvariant sein.

Das heißt, jede Instanz, in der sie aufgerufen wird, darf andere Instanzen nicht beeinflussen und globale Variablen, wie der Seed müssten in getrennten Speicherbereichen gehalten werden.

Aufgabe 5)

b)



Aufgabe 6)

- a) `printf()` gibt für jeden Thread den Wert 1 aus. Da `a=0` und `private` ist, wird es für jeden Thread neu inkrementieren.