Aufgabe 3

3.a:

A ist ein array der Länge n

$Function\, quickSort\,(A : Sequence\, of\; Element) : Sequence\; of\; Element$

$if\, |A| \leq 1\, then\, return\; A$

$pick\; p \in A\, ist\; eine\, Mediane\; des\; Arrays\; A$

$a := \langle e \in A : e > p \rangle$

$b := \langle e \in A : e > p \rangle$

$c := \langle e \in A : e > p \rangle$

$return\, concatenation(quickSort(a), b, quickSort(c))$

aus der Vorlesung ist bekannt, dass eine Mediane ein „Best Case“ für QuickSort ist.
Das Master-Theorem in diesem Fall sieht so aus:

$T(n) = 2\mathrm{T}(n/2) + O(n)$

$a=2$ $\quad 1 = \log_2(2)$

$b=2$ $\quad$ daraus folgt:

$d=1$ $\quad O(n \log(n))$

3.b:

in der Praxis wissen wir nichts über die Eingabe, deshalb wissen wir auch nich was eine Mediane ist und können auch nicht so einfach eine Mediane nehmen.

3.c:

A ist ein array der Länge n

$Function\, quickSort\,(A : Sequence\, of\; Element) : Sequence\; of\; Element$

$if\, |A| \leq 1\, then\, return\; A$

$pick\; p \in A\, ; P = A_x ; x = 1/n \sum_{n=0}^{n} A_n$

$a := \langle e \in A : e > p \rangle$

$b := \langle e \in A : e > p \rangle$

$c := \langle e \in A : e > p \rangle$

$return\, concatenation(quickSort(a), b, quickSort(c))$

in diesem Fall wissen wir ganz genau nicht was für ein Wert in dieser Zahl steht. Jetzt gehe ich davon aus, dass in dieser Zahl das letzte oder erste Element steht.

Das Master-Theorem sieht so aus:

$T(n) = 2\mathrm{T}(n/n-1) + O(n)$

$a = 2$ $\quad$ $1 \, ? \, 1/\log_2(n-1)$

$b = n-1$ $\quad$ daraus folgt:

$d = 1$ $\quad$ für $n=3$ gilt $1 = 1/\log_2(2)$ und die Laufzeit ist $O(n \log n)$

für $n > 3$ gilt $1 > 1/\log_2(2)$ die Laufzeit ist $O(n^2)$