Blatt-02

Mehmet Kaan Isik, mi84, 5360154

Andrés Fernández Lebrón, af231, 3757560

Paramie Willmann, pw221, 3534685

Aufgabe 1.

a.) $2n^3 + 4n^2 + 7\sqrt{n} \in O(n^3)$

Beweis $\quad 2n^3 + 4n^2 + 7\sqrt{n} \leq c \cdot g(n) \quad \forall\, n \geq n_0$

$2n^3 + 4n^2 + 7\sqrt{n} \leq c \cdot n^3$ $\quad$ sei $n = 1$

$$\frac{2n^3 + 4n^2 + 7\sqrt{n}}{n^3} \leq c$$

$$2 + \frac{4}{n} + \frac{7\sqrt{n}}{n^2} \leq c$$

$$2 + \frac{4}{1} + \frac{7\sqrt{1}}{1} \leq 13$$ ✓ Es existiert genau ein c

so $2n^3 + 4n^2 + 7\sqrt{n} \leq 13 \cdot g(n)$ ist wahr

b.) $n \cdot \log_3(n) \in \omega(n \cdot \log_5(n))$

Beweis $\quad n \cdot \log_3(n) \geq c \cdot g(n) \quad \forall\, n \geq n_0$

$n \cdot \log_3(n) \geq c \cdot n \cdot \log_5(n)$

$$\frac{\log_3 n^2}{\log_5 n^2} \geq c$$ $\quad$ sei $n = 1$

$$\frac{\log_3 1}{\log_5 1} \geq c$$

$$0 \geq c$$

dann $n \cdot \log_3(n) \in \omega(n \cdot \log_5(n))$ ist nicht wahr c.) $2^n \in o(n!)$

Beweis $2^n \leq c \cdot g(n) \quad \forall c > 0$

$2^n \leq c \cdot (n \cdot (n-1) \cdot (n-2) \dots) \quad \forall n \geq n_0$

$2^n \leq c \cdot (1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 \cdot \ldots \cdot n)$ sei $n = 1$

$2^1 \leq c \cdot (1)$

$2 \leq c$

$2^n \leq 2 \cdot (n!)$ ist wahr

$2^3 \leq 2 \cdot (3 \times 2 \times 1)$
$8 \leq 12$ ✓

$2^5 \leq 2 \cdot (5 \times 4 \times 3 \times 2 \times 1)$
$32 \leq 2 \cdot 120$
$32 \leq 240$ ✓

Es existiert genau ein C

d.) $2\log_2(n^2) \in \Omega((\log_2 n)^2)$

Beweis $2\log_2(n^2) \geq c \cdot g(n) \quad \forall n \geq n_0$

$2\log_2(n^2) \geq c \cdot (\log_2 n)^2$

$$\frac{4 \cancel{\log_2 n}}{(\cancel{\log_2} n) \log_2 n} \geq c$$

$\frac{4}{\log_2 n} \geq c$ sei $n = 1$

$\frac{4}{0} \geq c$ ist nicht wahr dann gibt es keine bestimmte Konstanz.

$2\log_2(n^2) \in \Omega((\log_2 n)^2)$ ist nicht wahr

e.) $\max\{f(n), g(n)\} \in \Theta(f(n)+g(n))$
für nicht negative Funktionen f und g.

Beweis $\quad c_1 \cdot g(n) \leq \max\{f(n), g(n)\} \leq c_2 \cdot g(n) \quad \forall\, n \geq n_0$

$$c_1 \cdot (f(n)+g(n)) \leq \max\{f(n), g(n)\} \leq c_2 \cdot (f(n)+g(n))$$

sei $c_1 = 1$
$c_2 = 2$

$$f(n)+g(n) \leq \max\{f(n), g(n)\} \leq 2(f(n)+g(n))$$

$$f(n)+g(n) \leq \max\{f(n), g(n)\} \leq 2f(n)+2g(n)$$

hier ist nicht wahr

(-1) das ist wahr wenn c_1 = 1/2

zz $\quad f(n)+g(n) \leq \max\{f(n), g(n)\}$ ist nicht wahr

Beweis $\quad$ sei $f(n) = 4n \quad$ für $n = 1$
$g(n) = 3n$

$$f(n)+g(n) \leq \max\{4n, 3n\}$$

$$4n + 3n \leq 4n$$

$$7n \not\leq 4n \quad \text{ist nicht wahr}$$

zz $\quad \max\{f(n), g(n)\} \leq 2f(n)+2g(n)$ ist wahr

Beweis $\quad$ sei $f(n) = 4n \quad$ für $n = 1$
$g(n) = 3n$

$$\max\{4n, 3n\} \leq f(n)+g(n)$$

$$4n \leq 4n + 3n$$

$$4n \leq 7n \quad ✓$$

Dann $\quad \max\{f(n), g(n)\} \in O(f(n)+g(n))$
für nicht negative Funktionen f und g.

Aufgabe 2.

$\sqrt{n}$ $< O$	2^n $= O$	$n!$ $> O$	$\log(n^3)$
3^n $> O$	n^{100} $> O$	$\log(\sqrt{n})$ $< O$	$(\log n)^2$
$\log n$ $< O$	$10^{100} \cdot n$ $= O$	$(n+1)!$ $> O$	$n \log n$
$2^{(n^2)}$ $< O$	n^n $> O$	$\sqrt{\log n}$ $< O$	$(2^n)^2$

$n^n > (n+1)! > n! > 2^{n^2} > (2^n)^2 > 3^n > 2^n > n^{100} > n \cdot log(n) > 10^{100}n >$
$(log(n))^2 > \sqrt{n} > log(n^3) > log(n) > log(\sqrt{n}) > \sqrt{log(n)}$

(-0.5) 2^(n^2) wächst asymptotisch am schnellsten(-0.5) (log(n))^2 und sqrt(n)sind v

Aufgabe 3.

a)

3	1	1	7	4	3	4	3
blue	green	red	gray	yellow	orange	white	black

sei A[0] = Pivot Dann (3, "blue") ist Pivot

alle Elemente die < Pivot werden am linken Seite des Arrays. auch die Elemente > Pivot werde am rechte Seite des Arrays.

=>

3	1 ✓	1 ✓	7 ✗	4	3	4	3 ✗
blue	green	red	gray	yellow	orange	white	black

swap

=>

3	1	1	3	✗ 4	✗ 3	✓ 4	7
blue	green	red	black	yellow	orange	white	gray

swap

=>

3	1	1	3	3	4	4	7
blue	green	red	black	orange	yellow	white	gray

swap

muss nicht mehr sortieren.

=>

3	1 ✓	1 ✓	3 ✗	3 ✗	4	4	7
black	green	red	blue	orange	yellow	white	gray

swap

=>

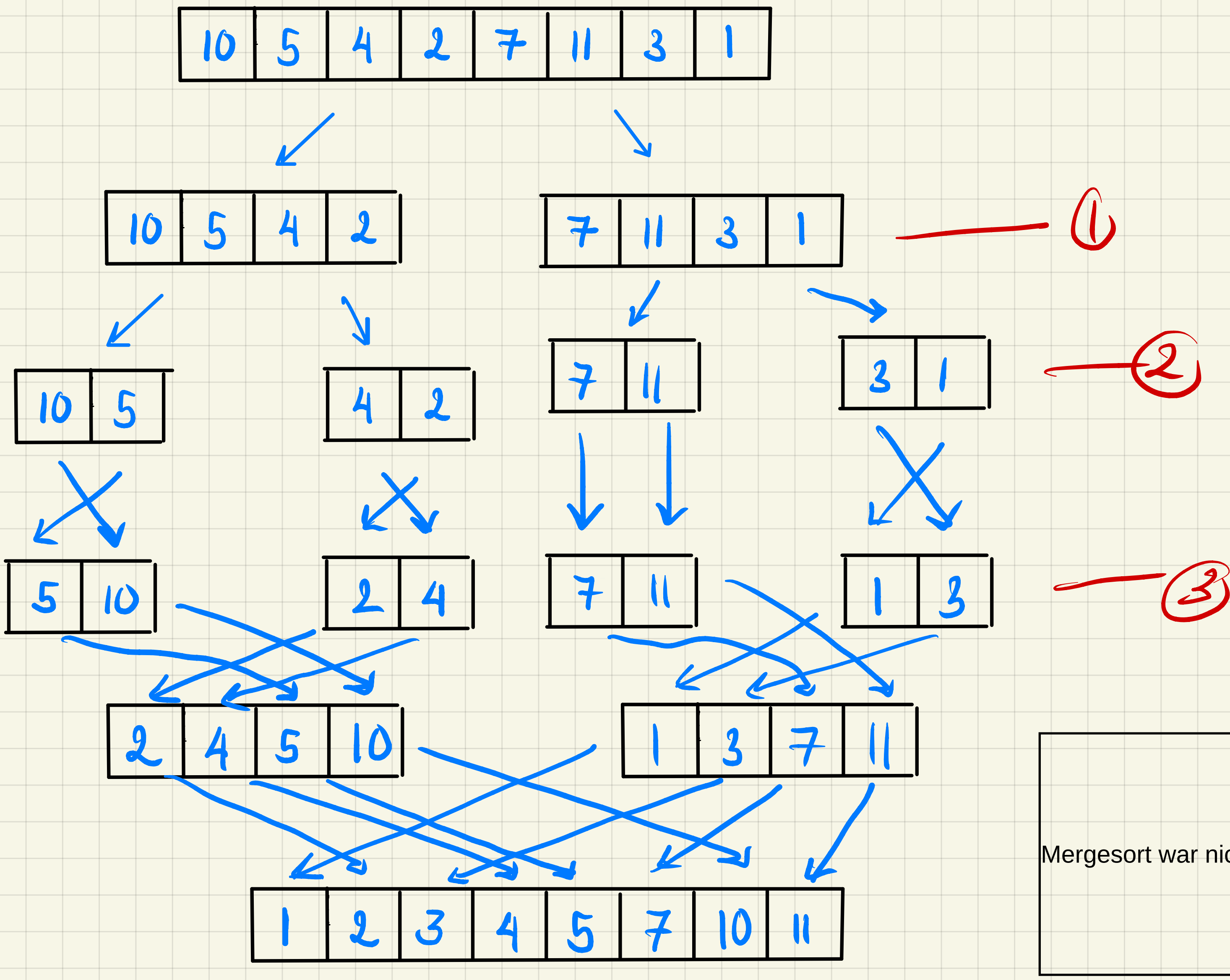
1	1	3	3	3	4	4	7
red	green	black	blue	orange	yellow	white	gray

mit dem Quicksort

b.) QuickSort → Worst case $O(n^2)$
→ Best case $O(n \log n)$

MergeSort ist stabil, weil alle Elemente, die gleiche Schlüssel haben, bleiben noch an der ursprüng Folge.

Die Laufzeit von MergeSort ist $T(n) \in O(n \cdot \log n)$



Mergesort war nich

Eine Methode ein vergleichsorientierter Sortieralgorithmus stabil zu machen, ist es, eine Variable zu erstellen, die die ursprünglichen Indexen der Elemente speichert und sortiert diese, indem sie die ursprünglichen Indexen vergleicht, nur wenn die Keys der Elementen gleich sind.

(-3) Begründung für die Asymptotische Laufzeit fehlt