aufgabe 7

gruppe NOFAL
Hattinger
Nicolic

① $f_1 + f_2 \in O(g)$ wahr weil

$$\boxed{\begin{array}{l} f_1 \in O(g) \Leftrightarrow \exists\, c_1 \in \mathbb{R}^+, n_0 \in \mathbb{N} \text{ sodass } f_1 \leq c_1 \cdot g(n) \\ f_2 \in O(g) \Leftrightarrow \exists\, c_2 \in \mathbb{R}^+, n_0 \in \mathbb{N} \text{ s. dass } f_2 \leq c_2 \cdot g(n) \end{array}}$$

$f_1 + f_2 \leq c_1\, g(n) + c_2\, g(n) = g(n) \cdot \underline{(c_1 + c_2)}$

$f_1 + f_2 \leq (c_1 + c_2)\, g(n) = c_3\, g(n)$ wobei $c_3 \geq c_1 + c_2 > 0$

$\Rightarrow f_1 + f_2 \leq c_3 \cdot g(n) \Leftrightarrow f_1 + f_2 \in O(g)$

② $f_1 - f_2 \in O(1)$ falsch weil

gegen Beispiel wenn $f_1 = n^3$, $f_2 = n^2$, $g = n^3$

$f_1 - f_2 = n^3 - n^2 \in n^3$

begründung $n^3 - n^2 \leq c \cdot n^3$ wenn $c = 1$, $n_0 = 2$

dann gilt $n^3 - n^2 \leq n^3 \quad \forall n \geq n_0$

Probe $8 - 4 \leq 8 \quad \forall n \geq 2$

$\Rightarrow f_1 - f_2 \in O(n^3) \Rightarrow f_1 - f_2 \in O(1)$

③ $f_1 \cdot f_2 \in O(g^2)$ wahr weil

$f_1 \cdot f_2 \leq c_1 \cdot g(n) \cdot c_2 \cdot g(n) = (c_1 \cdot c_2) \cdot g^2(n) = c_3\, g^2(n)$

wobei $c_3 \geq c_1 \cdot c_2 > 0$

$\Rightarrow f_1 \cdot f_2 \leq c_3\, g^2(n) \Leftrightarrow f_1 \cdot f_2 \in O(g^2)$

④ $\frac{f1}{f2} \in O(1)$ falsch weil

gegenbeispiel

Wenn $f1 = n^3$, $f2 = n^2$, $g = n^3$

$\Rightarrow \frac{f1}{f2} = \frac{n3}{n2} = n \in O(n)$

$\Rightarrow \frac{f1}{f2} \notin O(1)$

⑤ $f_1 \in O(f_2) \wedge f_2 \in O(f_1)$ falsch weil

gegenbeispiel wenn $f_1 = n^3$, $f_2 = n^2$, $g = n^3$

$$\left.\begin{array}{c} f_1 = n^3 \notin O(n^2) \\ \text{aber} \\ f_2 = n^2 \in O(n^3) \end{array}\right\} \Rightarrow \text{falsch}$$

⑥ in diesem fall haben wir 3 fälle zum untersuchen

- $f_1 < f_2$:- $f_1 \in O(f_2) \wedge f_2 \notin O(f_1)$
- $f_1 > f_2$:- $f_1 \notin O(f_2) \wedge f_2 \in O(f_1)$
- $f_1 = f_2$:- $f_1 \in O(f_2) \wedge f_2 \in O(f_1)$

$\Rightarrow$ wahr

- aufgabe (8)

Def von $O(f)$

(A) $O(f) := \{ g: N \to R^+ \mid \underline{\underline{\exists}}\ C > 0, \exists n_0 \in N, \forall n > n_0 : g(n) \leq c \cdot f(n) \}$

Def von $o(f)$

(B) $o(f) := \{ g: N \to R^+ \mid \forall\ \underline{\underline{C}} > 0, \exists n_0 \in N, \forall n > n_0 : g(n) \leq c \cdot f(n) \}$

oder $\lim\limits_{x \to \infty} \frac{g(x)}{f(x)} = 0 \iff g \in o(f)$

Nach der Def von $O(f)$:- bedeutet das, dass die menge von funktionen "g" Wachsen nicht wesentlich schneller als f aber sie dürfen auch so schnell wie f Wachsen

BSP $f = n^2 \in O(n^2)$

Nach der Def von $o(f)$:- bedeutet das, dass die menge von funktionen g Wachsen langsamer, (Müssen) langsamer Wachsen als f

BSP $f = n^2 \notin o(n^2)$ aber $f = n \in o(n^2)$

von der obigen def sehen wir bei $O(f)$ steht $\exists c$, also es reicht wenn wir nur eine constante $\underline{\underline{c}}$ finden können so dass diese beziehung gilt aber nach der def von ~~o(f)~~ ~~d~~ $o(f)$ diese beziehung muss gilten für alle constanten c

folgerung

wenn $f \in o(g) \Rightarrow f \in O(g)$

aber die umkehrung gilt nicht

BSP $f(n) = n \in o(n^2) \Rightarrow f(n) = n \in O(n^2)$

aber die umkehrung nicht :- $f(n) = n^2 \in O(n^2) \underset{nicht}{\not\Rightarrow} f(n) = n^2 \in o(n^2)$

aufgabe 9

- LinearSearch(A,v)

```
for i ← 1 to length[A]
        do if A[i]=v
              then return i
return 0
```

Loop invariante:- vor jeder loop iteration haben wir $A[j] \neq v$, $\forall\ j < i$

1- initialization:- vor der ersten loop iteration ist die invariante wahr

2- erhaltung:- die invariante wird bei jeder iteration erhalten (widerspruchs beweis)

annahme:- invariante wird nicht erhalten so dass bei der i-the iteration gibt es $j < i$ mit $A[j]=v$ aber in diesem fall hatte die j-the iteration den wert j schon zurückgegeben also es sollten keine i-the iteration geben, ~~[illegible]~~ widerspruch zur annahme $\Rightarrow$ invariante wird erhalten

3- Terminierung: wenn die iteration (schleife) terminiert dann gibt es 2 fälle:-

$\boxed{A}$ Terminierung wenn $i \leq length[A]$, i wird zurückgegeben wenn $A[i]=v$

$\boxed{B}$ Terminierung wenn $i > length[A]$, $A[j] \neq v$, 0 wird zurück gegeben.