3.3.

1. Abbruchbedingung: 1. element = 0

2. ackermann n 0: nach n > n - 1
   ↳ n läuft wieder 0

3. ackermann n m: n > n-1, m > m-1
   (laufen beide gegen 0)

↳ Abbruchbedingung wird erreicht

3.1 1. Abbruchbedingung 1 Element in Liste oder 0
   --> bubbleSort wird mit List rekursiv aufgerufen.
*1 D.h, die Liste wird immer mit einem Element weniger aufgerufen.
   ↳ bubbleSort terminiert

2. Abbruchbedingung bubble: 1 Element oder leere Liste

*2 --> bubble wird immer rekursiv mit der restliste aufgerufen, also 1 element weniger als vorher
   ↳ terminiert

*1 + *2 --> bubbleSort terminiert

3.2. minimum:

1. Abbruchbedingung: kürzeste Liste = 1
2. Liste verliert immer 1 Element
   (Liste vorher) < (Liste vorher - 1)
   ↳ Terminiert

loesche:

1. Abbruchbedingung ist leere Liste entspricht ✓
2. loesche wird im rekursiven Aufruf aufgerufen mit einem Element weniger als vorher.
   Liste davor < Liste davor - 1.
   - ↳ irgendwann wird leere Liste, oder das gesuchte Element erreicht
     ↳ terminiert

2. Abbruchbedingung ist leere Liste.

schlussfolgerung: 1. Abbruchbedingung ist leere Liste

2. Es wird das kürzeste El. genommen und vorn rausgeholt. Im rekursiven Aufruf dieses Element gelöscht und die Liste mit gekürztem Element aufgerufen. Hier gilt dann Liste < Liste-1.
   ↳ Abbruchbed. wird erreicht -> terminiert.

BRUNNEN

1.) 1. $T^{best}_{fun1}(n) = c_{(:)} = T^{worst}_{fun1}(n)$

$c_{(:)} = 1$

$\Rightarrow O(n)$

- hängt Element vorne an die Liste

2. $T^{best}_{fun2}(0) = 1$

$T^{worst}_{fun2}(n) = c_{(:)} + T^{worst}_{fun2}(n-1)$

$= n \cdot c_{(:)} + T^{worst}_{fun2}(c_{(:)}) = n+1 = T^{best}_{fun2}(n)$

$c_{(:)} = 1$

$\Rightarrow O(n)$

- hängt Element am Ende der Liste an

3. $T^{best}_{fun3}(x,y) = c_{(>)} + c_{(>)} + c_{(==)} = 3$

$T^{worst}_{fun3}(x,y) = 3 \cdot c_{(>)} + T^{worst}_{fun3}((x-y),y)$

$f(-) = 1$

$= 4 \cdot \frac{x}{y}$

$\Rightarrow O(n)$

- reduziert solange $x-y$, bis $x < y$

4. $T^{worst}_{fun4}(n) = 4 \cdot \frac{n}{2} + c_{(==2)}$

$c_{(==2)} = 1$

$= 4 \frac{n}{2} + 1 = T^{best}_{fun4}$

$\Rightarrow O(n)$

- ruft Funktion 3 mit 2 als y-Wert auf und prüft, ob dieser 0 ist

5. $T^{best}_{fun5}(0) = 1$

$T^{worst}_{fun5}(n) = c_{(:)} + T^{worst}_{fun5}(n-1) + c_{(-)}$

$= n \cdot c_{(:)} + n \cdot c_{(-)} + T^{worst}_{fun5}(n-1)$

$= n + 2$

$\Rightarrow O(n)$

- verwirft die Elemente nach eingegebener Zahl

6. $T^{best}_{fun6}(0, m) = 1$

$T^{best}_{fun6}(n, 0) = 1 \quad T^{best}_{fun6} = 1$

$T^{worst}_{fun6}(m, n) = 1 + 1 + T^{worst}_{fun6}(n-1, n-1)$

$= 2 + \left(\frac{n+m}{2}\right)$

$= 2 + \frac{n}{2} + \frac{m}{2}$

$\Rightarrow O(n)$

- bildet Tupel, bis eine Liste leer ist.

7. $T^{best}_{fun7}(1) = 1$

$T^{worst}_{fun7}(n) = T_{fun7}(n-1) + n \cdot c_{(>)} + n$

$= 3n$

$\Rightarrow O(n)$

- gibt höchstes Element der Liste zurück

8. $T^{best}_{fun8}(0) = 1$

$T^{worst}_{fun8}(n) = n \cdot c_{(=2)} + n \cdot c_{(1)} + T(n-1)$

$= 2n + 1$

$\Rightarrow O(n)$

- löscht Element aus Liste

9. $T^{best}_{fun9}(n) = 1 + 2 + T^{best}_{fun7}(n) + T^{best}_{fun8}(n)$

$= 1 + 1 + 1$

$= 3$

$T^{worst}_{fun9}(n) = 1 + c_{(=)} \cdot n + (T^{worst}_{fun7}(n) + T^{worst}_{fun8}(n))$

$= (3n + 2n + 1)$

$= 5n^2 + n$

$\Rightarrow O(n^2)$

- ordnet die Liste aufsteigend

2.) 1)

a) $(n+1)^2 \in O(n^2)$

$(n^2+2n+1) \in O(n^2)$?

$O(n^2+2n+1) = O(n^2) + O(2n) + O(1)$

$O(1) \in O(n^2)$ ✓, $O(n^2) \in O(n^2)$ ✓

$O(2n) \in O(n^2) \Rightarrow O(2) + O(n) = O(n)$

~~falsch~~

$\Rightarrow O(2) \in O(1) \in O(n)$

b) $\log(n) + n^2 + n^4 \in O(n^5)$

$O(\log(n) + n^2 + n^4) = O(\log(n)) + O(n^2) + O(n^4)$

$O(n^4) \in O(n^5)$ ✓

$O(n^2) \in O(n^5)$ ✓

$O(\log(n)) \in O(n) < O(n^5)$

$\Rightarrow O(\log(n)) \in O(n^5)$

$\log(n) \leq c \cdot n^5$

$\lim_{n \to \infty} \frac{\log(n)}{n^5} = 0$

$\Rightarrow O(\log(n)) \in O(n^5)$

$\Rightarrow O(\log(n) + n^2 + n^4) \in O(n^5)$

c) $3n \cdot \log(n) + 7n \in O(n \cdot \log n)$

$3n \cdot \log(n) + 7n \leq c \cdot n \cdot \log(n) \quad | : (n \cdot \log(n))$

~~$\frac{3n \cdot \log(n) + 7n}{n \cdot \log(n)} \in c$~~

$\lim_{n \to \infty} \left(3n \log(n) + \frac{7n \log(n)}{n} \right) = \infty$ ↯

läuft also nicht gegen 0, Element

d) $|\sin(n)| \in O(1)$

da Sin nur die Werte -1 bis 1 annehmen kann, und der Betrag um ihn sin ist, ist es auch immer $|\sin(n)| \leq 1$

$\hookrightarrow |\sin(n)| \in O(1)$

e) $3^n \in O(2^n)$

$3^n \leq c \cdot 2^n$

$\frac{3^n}{2^n} \leq c$

$\left(\frac{3}{2}\right)^n \leq c$

$n \cdot \log\left(\frac{3}{2}\right) \leq \log(c)$

$n \leq \frac{\log(c)}{\log\left(\frac{3}{2}\right)}$

∞

$c, \infty \lightning \quad 3^n \notin O(2^n)$

f) $3^n \in 2^{O(n)}$

$3^{O(n)} \notin 2^{O(n)}$

da $3 > 2$

2. $21 \leq \log n \leq \sqrt{n} \leq n \leq n \log n \leq n^2 \log n \leq n^2 \leq n^3 \leq e^n$