Aufgabe 1. Zu zeigen ist

$$f(n) = \mathrm{O}(g(n)) \wedge g(n) = \mathrm{O}(h(n)) \Rightarrow f(n) = \mathrm{O}(h(n)).$$

Angenommen die linke Seite der Implikation,

$$\exists\, c > 0 \;\exists\, n_0 \in \mathbb{N} : \forall\, n \geq n_0 : |f(n)| \leq c|g(n)| \tag{1}$$
$$\exists\, k > 0 \;\exists\, n_1 \in \mathbb{N} : \forall\, n \geq n_1 : |g(n)| \leq k|h(n)| \tag{2}$$

gilt. Die Ungleichung aus (1) kann zu $\frac{|f(n)|}{c} \leq |g(n)|$ umformuliert werden und in (2) eingesetzt werden:

$$\exists\, k > 0 \;\exists\, n_1 \in \mathbb{N} : \frac{|f(n)|}{c} \leq k|h(n)|$$
$$\exists\, k > 0 \;\exists\, n_1 \in \mathbb{N} : |f(n)| \leq ck|h(n)|$$

Dann gilt

$$\exists\, j > 0 \;\exists\, n_2 \in \mathbb{N} : \forall\, n \geq n_2 : |f(n)| \leq j|h(n)|$$

für $j = ck$ und $n_2 = \max\{n_0, n_1\}$. Somit gilt $f(n) = O(h(n))$ wenn $f(n) = O(g(n))$ und $g(n) = O(h(n))$, die O-Notation ist transitiv.

Aufgabe 2. Es gilt $f(n) = \sqrt{n} = \Theta(\sqrt{n})$.

a) Es gilt $a = 1$, $b = 2$ und $c = \log_2(1) = 0$. Es greift Fall 4 wegen $f(n) = \Omega(n^{0+\frac{1}{2}}) = \Omega(\sqrt{n})$ ($\epsilon = \frac{1}{2}$). Somit gilt $T(n) = \Omega(\sqrt{n})$.

b) Es gilt $a = 2$, $b = 2$ und $c = \log_2(2) = 1$. Es greift Fall 1 wegen $f(n) = \mathrm{O}(n^{1-\frac{1}{2}}) = \mathrm{O}(\sqrt{n})$ ($\epsilon = \frac{1}{2}$). Somit gilt $T(n) = \Theta(\sqrt{n})$.

c) Es gilt $a = 2$, $b = 4$ und $c = \log_4(2) = \frac{1}{2}$. Es greifen Fall 2 und 3 wegen $f(n) = \mathrm{O}(n^{\frac{1}{2}} \log(n)^0) = \mathrm{O}(\sqrt{n})$ und $f(n) = \Omega(n^{\frac{1}{2}} \log(n)^0) = \Omega(\sqrt{n})$ ($k = 0$). Somit gilt $T(n) = \Theta(\sqrt{n}\log(n))$.

Aufgabe 3. Die folgende Tabelle zeigt für jede Zeile die maximale Anzahl der Ausführungen („Häufigkeit") und die daraus resultierende maximale Anzahl an Operationen. Nachdem etwa Zeile 3 zu einem früheren Abbruch von Schleifeniterationen führen kann sind diese Werte als oberes Limit zu verstehen, nicht zwingend als tatsächlich erreichbares Maximum.

Zeile	Häufigkeit	Operationen	
1	1	0	
2	1	n	
3	n	$3n$	
4	n	$2n^2$	Größtmögl. $i = n$ also $2n \cdot n = 2n^2$
5	$2n^2$	$12n^2$	
6	$2n^2$	$6n^3 + 1$	Größtmögl. $j = 2i = 2n$ also $2n + n = 3n$ und $2n^2 3n = 6n^3$
7	$6n^3$	0	
8	$6n^3$	$18n^3$	
9	$6n^3$	0	
10	$6n^3$	$24n^3$	
11	1	0	

Die größte vorkommende maximale Anzahl der Operationen ist $24n^3$, somit benötigt der Algorithmus $O(n^3)$ arithmetische Operationen.