Algorithms: Solutions 4

Problem 1
Give an example of functions $f(n)$ and $g(n)$ that satisfy all of the following conditions:

$f(n) = O(g(n))$
$f(n) \neq \Theta(g(n))$
$f(n) \neq o(g(n))$

Consider the following two functions:

$$\begin{aligned} f(n) &= 1 \\ g(n) &= \begin{cases} 1 & \text{if } n \text{ is even;} \\ n & \text{if } n \text{ is odd.} \end{cases} \end{aligned}$$

Since $f(n) \leq g(n)$, we immediately conclude that $f(n) = O(g(n))$. For even n, the function $f(n)$ is of the same order as $g(n)$, which means that $f(n) \neq o(g(n))$. On the other hand, for odd n, $f(n)$ grows asymptotically slower than $g(n)$, which implies that $f(n) \neq \Theta(g(n))$.

Problem 2
Give a precise mathematical proof of the following asymptotic bounds:

(a) $\sqrt{n} = o(n)$

We need to show that, for every $c > 0$, there is some n_0 such that, for all $n \geq n_0$, we have $\sqrt{n} < c \cdot n$. We define n_0 as follows:

$$n_0 = \left\lceil \frac{1}{c^2} + 1 \right\rceil.$$

Then, for every $n \geq n_0$, we have $n > 1/c^2$, which implies that $\sqrt{n} \cdot c > 1$ and readily leads to the desired inequality:

$$\sqrt{n} < \sqrt{n} \cdot (\sqrt{n} \cdot c) = n \cdot c.$$

(b) $(n+1)^a = \Theta(n^a)$

If $n \geq 1$, then

$$(n+1)^a \leq (2n)^a = 2^a \cdot n^a.$$

Thus, we get the following bounds for $(n+1)^a$:

$$n^a \leq (n+1)^a \leq 2^a \cdot n^a,$$

which implies that $(n+1)^a = \Theta(n^a)$.