1 Exercise 1

1.1 a

To show: $2n^2 + 10n - 5 = \Theta(n^2)$

Without abuse of notation, we want to show that $2n^2 + 10n - 5 \in \Theta(n^2)$

By definition, $f(n) \in \Theta(g(n))$ if $f(n) \in O(g(n))$ and $f(n) \in \Omega(g(n))$

So, let's show that $2n^2 + 10n - 5 \in O(n^2)$ first:

Definition of $f(n) \in O(g(n))$ is $\exists c > 0, n_0 \in \mathbb{N}$ s.t. $\forall n > n_0, f(n) \leq c \cdot g(n)$

Let $c = 3$, then $2n^2 + 10n - 5 \leq 3n^2$ for all $n \geq 1$(our n_0) $\in \mathbb{N}$

Now, let's show that $2n^2 + 10n - 5 \in \Omega(n^2)$:

Definition of $f(n) \in \Omega(g(n))$ is $\exists c > 0, n_0 \in \mathbb{N}$ s.t. $\forall n > n_0, f(n) \geq c \cdot g(n)$

Let $c = 1$, then $2n^2 + 10n - 5 \geq n^2$ for all $n \geq 1$(our n_0) $\in \mathbb{N}$

We have shown that $2n^2 + 10n - 5 \in O(n^2)$ and $2n^2 + 10n - 5 \in \Omega(n^2)$, so $2n^2 + 10n - 5 \in \Theta(n^2)$.

1.2 b

To show: $2^{2n} = \omega(2^n)$

Without abuse of notation, we want to show that $2^{2n} \in \omega(2^n)$

By definition, $f(n) \in \omega(g(n))$ if $\forall c > 0, \exists n_0 \in \mathbb{N}$ s.t. $\forall n > n_0, f(n) > c \cdot g(n)$.

Let $c = 1$, then $2^{2n} > 2^n$ for all $n \geq 1$(our n_0) $\in \mathbb{N}$

We have shown that $2^{2n} = \omega(2^n)$.

1.3 c

To show: $o(n) \subset O(n^2)$

Let $f(n) \in o(n)$, we show that $f(n) \in O(n^2)$

By definition, $f(n) \in o(g(n))$ if $\forall c > 0, \exists n_0 \in \mathbb{N}$ s.t. $\forall n > n_0, f(n) < c \cdot g(n)$

We need to show that $f(n) \in O(n^2)$, which means $\exists c > 0, n_0 \in \mathbb{N}$ s.t. $\forall n >$

$n_0, f(n) \leq c \cdot n^2$

Let $c = 1$, then by definition of $f(n) \in o(n)$, $\exists n_0 \in \mathbb{N}$ s.t. $\forall n > n_0, f(n) < n$

Now, $f(n) < n < n^2$ for all $n \geq n_0$, so $f(n) \in O(n^2)$

1.4 d

We want to show that for arbitrary functions $f : \mathbb{N} \to \mathbb{N}$ and $g : \mathbb{N} \to \mathbb{N}$, the following holds:

$$f(n) \in o(g(n)) \Leftrightarrow g(n) \in \omega(f(n))$$

Direction 1: $f(n) \in o(g(n)) \Rightarrow g(n) \in \omega(f(n))$:

By definition of $f(n) \in o(g(n))$, $\forall c > 0, \exists n_0 \in \mathbb{N}$ s.t. $\forall n > n_0, f(n) \leq c \cdot g(n)$

Now, let $c > 0$ be arbitrary. Then, given n_0 from the definition of $f(n) \in o(g(n))$, we have that $\forall n > n_0, f(n) \leq c \cdot g(n)$.

This means that $g(n) \geq \frac{1}{c} \cdot f(n)$ for all $n > n_0$, so $g(n) \in \omega(f(n))$.

Direction 2: $g(n) \in \omega(f(n)) \Rightarrow f(n) \in o(g(n))$:

By definition of $g(n) \in \omega(f(n))$, $\forall c > 0, \exists n_0 \in \mathbb{N}$ s.t. $\forall n > n_0, g(n) \geq c \cdot f(n)$

Now, let $c > 0$ be arbitrary. Then, given n_0 from the definition of $g(n) \in \omega(f(n))$, we have that $\forall n > n_0, g(n) \geq c \cdot f(n)$.

This means that $f(n) \leq \frac{1}{c} \cdot g(n)$ for all $n > n_0$, so $f(n) \in o(g(n))$.

We have shown both directions, so $f(n) \in o(g(n)) \Leftrightarrow g(n) \in \omega(f(n))$.

2 Exercise 2

2.1 a

To show: $\forall c \in \mathbb{N} : c \cdot f(n) \in O(f(n))$

Let $c \in \mathbb{N}$ be arbitrary. We want to show that $c \cdot f(n) \in O(f(n))$.

By definition of $f(n) \in O(g(n))$, $\exists c' > 0, n_0 \in \mathbb{N}$ s.t. $\forall n > n_0, f(n) \leq c' \cdot g(n)$

Let $c' = c$, then $c \cdot f(n) \leq c \cdot f(n)$ for all $n \geq 1(\text{our } n_0) \in \mathbb{N}$

We have shown that $\forall c \in \mathbb{N} : c \cdot f(n) \in O(f(n))$.

2.2 b

To show: $f(n) + g(n) \in \Omega(f(n))$

By definition of $f(n) + g(n) \in \Omega(f(n))$, $\exists c > 0, n_0 \in \mathbb{N}$ s.t. $\forall n > n_0, f(n) + g(n) \geq c \cdot f(n)$

Let $c = 1$, then $f(n) + g(n) \geq f(n)$ for all $n \geq 1$(our n_0) $\in \mathbb{N}$. The important thing to note is that $g(n) \geq 0$.

We have shown that $f(n) + g(n) \in \Omega(f(n))$.

2.3 c

To show: $g(n) \in O(f(n)) \Rightarrow f(n) + g(n) \in O(f(n))$

By definition of $g(n) \in O(f(n))$, $\exists c > 0, n_0 \in \mathbb{N}$ s.t. $\forall n > n_0, g(n) \leq c \cdot f(n)$

We want to show that $f(n) + g(n) \in O(f(n))$, which means $\exists c' > 0, n_0 \in \mathbb{N}$ s.t. $\forall n > n_0, f(n) + g(n) \leq c' \cdot f(n)$

Let $c' = 2c$, then $f(n) + g(n) \leq 2c \cdot f(n)$ for all $n \geq 1$(our n_0) $\in \mathbb{N}$

We have shown that $g(n) \in O(f(n)) \Rightarrow f(n) + g(n) \in O(f(n))$.

2.4 d

To show: $O(f(n)) \cdot O(g(n)) = O(f(n) \cdot g(n))$

By definition of $O(f(n)) \cdot O(g(n)) = O(f(n) \cdot g(n))$, $\exists c_1, c_2 > 0, n_0 \in \mathbb{N}$ s.t. $\forall n > n_0, f(n) \leq c_1 \cdot f(n)$ and $g(n) \leq c_2 \cdot g(n)$

We want to show that $f(n) \cdot g(n) \leq c \cdot f(n) \cdot g(n)$ for all $n > n_0$.

Let $c = c_1 \cdot c_2$, then $f(n) \cdot g(n) \leq c_1 \cdot f(n) \cdot c_2 \cdot g(n)$ for all $n > n_0$

We have shown that $O(f(n)) \cdot O(g(n)) = O(f(n) \cdot g(n))$.

3 Exercise 3

Let c_i be the cost of the ith operation. We want to come up with a constant amortized cost of the operations.

Let D_i be the data structure after the ith operation, and let $\Phi(D_i)$ be the potential of the data structure after the ith operation.

Let $\Phi(D_i) = i - 2^{\lfloor \log_2 i \rfloor}$ be the potential function.

Notice, for $i, i+1$, if neither i nor $i+1$ are powers of 2, then $\lfloor \log_2 i \rfloor = \lfloor \log_2(i+1) \rfloor$, so $\Phi(D_{i+1}) = \Phi(D_i) + 1$.

If $i+1$ is a power of 2, then $\lfloor \log_2(i+1) \rfloor = \lfloor \log_2 i \rfloor + 1$, so

$$\begin{aligned}
\Phi(D_{i+1}) &= (i+1) - 2^{\lfloor \log_2(i+1) \rfloor} \\
&= i + 1 - 2^{\lfloor \log_2 i \rfloor + 1} \\
&= i + 1 - 2 \cdot 2^{\lfloor \log_2 i \rfloor} \\
&= i - 2^{\lfloor \log_2 i \rfloor} + (1 - 2^{\lfloor \log_2 i \rfloor}) \\
&= \Phi(D_i) + 1 - 2^{\lfloor \log_2 i \rfloor}
\end{aligned}$$

If i is a power of 2, then the same case applies as when neither i nor $i+1$ are powers of 2.

So

$$\Phi(D_{i+1}) - \Phi(D_i) = \begin{cases} 1 - 2^{\lfloor \log_2 i \rfloor} = 1 - i & \text{if } i \text{ is a power of 2} \\ 1 & \text{otherwise} \end{cases}$$

Now, we want to show that the amortized cost of the operations is constant.

Let c_i be the cost of the ith operation, and let a_i be the amortized cost of the ith operation.

Then, $a_i = c_i + \Phi(D_i) - \Phi(D_{i-1})$.

We know that

$$c_i = \begin{cases} i & \text{if } i \text{ is a power of 2} \\ 1 & \text{otherwise} \end{cases}$$

Now

$$a_i = \begin{cases} i + 1 - i = 1 & \text{if } i \text{ is a power of 2} \\ 1 + 1 = 2 & \text{otherwise} \end{cases}$$

.

Since a_i is bound by constant 2, the amortized cost of the operations is constant.