Section 8.8
Recursive Definitions

Arithmetic Progressions

- Recall from section 2.4 that an arithmetic progression is a sequence of the form:

$$a, a + d, a + 2d, \cdots a + nd, \cdots$$

- For example, the series:

$$-1, 3, 7, 11, \cdots$$

is an arithmetic progression where $a = -1$ and $d = 4$

Arithmetic Progressions

- For the arithmetic progression:

$$-1, 3, 7, 11, \cdots$$

- Given any element of the sequence, it is a simple matter to produce the next element in the sequence by adding 4 to it
- This rule can be used to calculate any element of the sequence except for the first element: -1

Recursive Functions

- We can then create a function to compute the nth element of

$$-1, 3, 7, 11, \cdots$$

by giving two rules:

1. $f(n) = -1$ if n = 0
2. $f(n) = f(n-1) + 4$ if n > 0

Note that the definition of f uses f. This is called recursion, and f is a recursive function

Recursive Functions

- Instead of using $n > 0$ as a case for an argument of f, it is common to express the argument as $n + 1$, because if n is a natural number, then $n + 1 > 0$

$f(0) = -1$	Base case
$f(n + 1) = 4 + f(n)$	Recursive case

Recursive Functions

- Examples of computing using the recursive function f

$$f(0) \ = -1$$

Recursive Functions

- Examples of computing using the recursive function f

$$f(0) \ = -1$$

$$f(1)$$

Recursive Functions

- Examples of computing using the recursive function f

$$f(0) \;= -1$$

$$f(1) \;= 4 + f(0)$$

Recursive Functions

- Examples of computing using the recursive function f

$$f(0) = -1$$

$$\begin{aligned} f(1) &= 4 + f(0) \\ &= 4 + -1 \end{aligned}$$

Recursive Functions

- Examples of computing using the recursive function f

$$f(0) = -1$$

$$\begin{aligned} f(1) &= 4 + f(0) \\ &= 4 + -1 \\ &= 3 \end{aligned}$$

Recursive Functions

- Examples of computing using the recursive function f

$$f(2)$$

Recursive Functions

- Examples of computing using the recursive function f

$$f(2) = 4 + f(1)$$

Recursive Functions

- Examples of computing using the recursive function f

$$
\begin{aligned}
f(2) &= 4 + f(1) \\
&= 4 + 4 + f(0)
\end{aligned}
$$

Recursive Functions

- Examples of computing using the recursive function f

$$
\begin{aligned}
f(2) &= 4 + f(1) \\
&= 4 + 4 + f(0) \\
&= 4 + 4 + -1
\end{aligned}
$$

Recursive Functions

- Examples of computing using the recursive function f

$$
\begin{aligned}
f(2) &= 4 + f(1) \\
&= 4 + 4 + f(0) \\
&= 4 + 4 + -1 \\
&= 7
\end{aligned}
$$

Recursive Functions

- Examples of computing the recursive function f

$$f(5) \ = 4 + f(4)$$

Recursive Functions

- Examples of computing the recursive function f

$$
\begin{aligned}
f(5) &= 4 + f(4) \\
&= 4 + 4 + f(3)
\end{aligned}
$$

Recursive Functions

- Examples of computing the recursive function f

$$\begin{aligned} f(5) &= 4 + f(4) \\ &= 4 + 4 + f(3) \\ &= 4 + 4 + 4 + f(2) \end{aligned}$$

Recursive Functions

- Examples of computing the recursive function f

$$
\begin{aligned}
f(5) &= 4 + f(4) \\
&= 4 + 4 + f(3) \\
&= 4 + 4 + 4 + f(2) \\
&= 4 + 4 + 4 + 4 + f(1)
\end{aligned}
$$

Recursive Functions

- Examples of computing the recursive function f

$$\begin{aligned} f(5) &= 4 + f(4) \\ &= 4 + 4 + f(3) \\ &= 4 + 4 + 4 + f(2) \\ &= 4 + 4 + 4 + 4 + f(1) \\ &= 4 + 4 + 4 + 4 + 4 + f(0) \end{aligned}$$

Recursive Functions

- Examples of computing the recursive function f

$$
\begin{aligned}
f(5) &= 4 + f(4) \\
&= 4 + 4 + f(3) \\
&= 4 + 4 + 4 + f(2) \\
&= 4 + 4 + 4 + 4 + f(1) \\
&= 4 + 4 + 4 + 4 + 4 + f(0) \\
&= 4 + 4 + 4 + 4 + 4 + -1
\end{aligned}
$$

Recursive Functions

- Examples of computing the recursive function f

$$
\begin{aligned}
f(5) &= 4 + f(4) \\
&= 4 + 4 + f(3) \\
&= 4 + 4 + 4 + f(2) \\
&= 4 + 4 + 4 + 4 + f(1) \\
&= 4 + 4 + 4 + 4 + 4 + f(0) \\
&= 4 + 4 + 4 + 4 + 4 + -1 \\
&= 19
\end{aligned}
$$

Recursive Functions

- Example 2: Define a recursive function that computes a^n where a is a real number and n is a natural number
- The underlying sequence is

$$a^0, a^1, a^2, \cdots$$

- Given any number in the sequence, multiply it by a to get the next number in the sequence
- The first number in the sequence is 1

Recursive Functions

- Example 2: Define a recursive function that computes a^n where a is a real number and n is a natural number

$$f(0) = 1$$

$$f(n+1) = a \cdot f(n)$$

Recursive Functions

- Example 2: Define a recursive function that computes a^n where a is a real number and n is a natural number

$$f(4) \ = a \cdot f(3)$$

Recursive Functions

- Example 2: Define a recursive function that computes a^n where a is a real number and n is a natural number

$$
\begin{aligned}
f(4) &= a \cdot f(3) \\
&= a \cdot a \cdot f(2)
\end{aligned}
$$

Recursive Functions

- Example 2: Define a recursive function that computes a^n where a is a real number and n is a natural number

$$
\begin{aligned}
f(4) &= a \cdot f(3) \\
&= a \cdot a \cdot f(2) \\
&= a \cdot a \cdot a \cdot f(1)
\end{aligned}
$$

Recursive Functions

- Example 2: Define a recursive function that computes a^n where a is a real number and n is a natural number

$$
\begin{aligned}
f(4) &= a \cdot f(3) \\
&= a \cdot a \cdot f(2) \\
&= a \cdot a \cdot a \cdot f(1) \\
&= a \cdot a \cdot a \cdot a \cdot f(0)
\end{aligned}
$$

Recursive Functions

- Example 2: Define a recursive function that computes a^n where a is a real number and n is a natural number

$$
\begin{aligned}
f(4) &= a \cdot f(3) \\
&= a \cdot a \cdot f(2) \\
&= a \cdot a \cdot a \cdot f(1) \\
&= a \cdot a \cdot a \cdot a \cdot f(0) \\
&= a \cdot a \cdot a \cdot a \cdot 1
\end{aligned}
$$

Recursive Functions

- Example 2: Define a recursive function that computes a^n where a is a real number and n is a natural number

$$
\begin{aligned}
f(4) &= a \cdot f(3) \\
&= a \cdot a \cdot f(2) \\
&= a \cdot a \cdot a \cdot f(1) \\
&= a \cdot a \cdot a \cdot a \cdot f(0) \\
&= a \cdot a \cdot a \cdot a \cdot 1 \\
&= a^4
\end{aligned}
$$

Recursive Functions

- Example 3: Define a recursive function that computes $\sum_{k=0}^{n} k$ where n is a natural number
- The underlying sequence of sums is

$$0, 0 + 1, 0 + 1 + 2, \cdots$$

- Given the nth sum in the sequence, add $n + 1$ to get the next sum in the sequence
- The first, 0^{th}, sum in the sequence is 0

Recursive Functions

- Example 3: Define a recursive function that computes $\sum_{k=0}^{n} k$ where n is a natural number

$$f(0) = 0$$

$$f(n+1) = n + 1 + f(n)$$

Recursive Functions

- Example 3: Define a recursive function that computes $\sum_{k=0}^{n} k$ where n is a natural number

$$f(4) \; = 4 + f(3)$$

Recursive Functions

- Example 3: Define a recursive function that computes $\sum_{k=0}^{n} k$ where n is a natural number

$$
\begin{aligned}
f(4) &= 4 + f(3) \\
&= 4 + 3 + f(2)
\end{aligned}
$$

Recursive Functions

- Example 3: Define a recursive function that computes $\sum_{k=0}^{n} k$ where n is a natural number

$$\begin{aligned} f(4) &= 4 + f(3) \\ &= 4 + 3 + f(2) \\ &= 4 + 3 + 2 + f(1) \end{aligned}$$

Recursive Functions

- Example 3: Define a recursive function that computes $\sum_{k=0}^{n} k$ where n is a natural number

$$
\begin{aligned}
f(4) &= 4 + f(3) \\
&= 4 + 3 + f(2) \\
&= 4 + 3 + 2 + f(1) \\
&= 4 + 3 + 2 + 1 + f(0)
\end{aligned}
$$

Recursive Functions

- Example 3: Define a recursive function that computes $\sum_{k=0}^{n} k$ where n is a natural number

$$\begin{aligned} f(4) &= 4 + f(3) \\ &= 4 + 3 + f(2) \\ &= 4 + 3 + 2 + f(1) \\ &= 4 + 3 + 2 + 1 + f(0) \\ &= 4 + 3 + 2 + 1 + 0 \end{aligned}$$

Recursive Functions

- Example 3: Define a recursive function that computes $\sum_{k=0}^{n} k$ where n is a natural number

$$
\begin{aligned}
f(4) &= 4 + f(3) \\
&= 4 + 3 + f(2) \\
&= 4 + 3 + 2 + f(1) \\
&= 4 + 3 + 2 + 1 + f(0) \\
&= 4 + 3 + 2 + 1 + 0 \\
&= 10
\end{aligned}
$$

Recursive Functions

- Example 4: Define a recursive function that computes n factorial: $n! = 1 \cdot 2 \cdot \ldots \cdot n$ where n is a natural number
- Note that $0! = 1$
- The underlying sequence of products is:

$$1, \qquad 1 \cdot 1, \qquad 1 \cdot 1 \cdot 2, \qquad 1 \cdot 1 \cdot 2 \cdot 3, \qquad \cdots$$

- Given the nth product in the sequence, multiply it by $n + 1$ to get the next product in the sequence
- The first, 0^{th}, product in the sequence is 1

Recursive Functions

- Example 4: Define a recursive function that computes n factorial: $n! = 1 \cdot 2 \cdot \ldots \cdot n$ where n is a natural number

$$f(0) = 1$$

$$f(n+1) = f(n) \cdot (n+1)$$

Recursive Functions

- Example 4: Define a recursive function that computes n factorial: $n! = 1 \cdot 2 \cdot \ldots \cdot n$ where n is a natural number

$$f(4) \ = f(3) \cdot 4$$

Recursive Functions

- Example 4: Define a recursive function that computes n factorial: $n! = 1 \cdot 2 \cdot \ldots \cdot n$ where n is a natural number

$$\begin{aligned} f(4) &= f(3) \cdot 4 \\ &= f(2) \cdot 3 \cdot 4 \end{aligned}$$

Recursive Functions

- Example 4: Define a recursive function that computes n factorial: $n! = 1 \cdot 2 \cdot \ldots \cdot n$ where n is a natural number

$$
\begin{aligned}
f(4) &= f(3) \cdot 4 \\
&= f(2) \cdot 3 \cdot 4 \\
&= f(1) \cdot 2 \cdot 3 \cdot 4
\end{aligned}
$$

Recursive Functions

- Example 4: Define a recursive function that computes n factorial: $n! = 1 \cdot 2 \cdot \ldots \cdot n$ where n is a natural number

$$\begin{aligned} f(4) &= f(3) \cdot 4 \\ &= f(2) \cdot 3 \cdot 4 \\ &= f(1) \cdot 2 \cdot 3 \cdot 4 \\ &= f(0) \cdot 1 \cdot 2 \cdot 3 \cdot 4 \end{aligned}$$

Recursive Functions

- Example 4: Define a recursive function that computes n factorial: $n! = 1 \cdot 2 \cdot \ldots \cdot n$ where n is a natural number

$$\begin{aligned} f(4) &= f(3) \cdot 4 \\ &= f(2) \cdot 3 \cdot 4 \\ &= f(1) \cdot 2 \cdot 3 \cdot 4 \\ &= f(0) \cdot 1 \cdot 2 \cdot 3 \cdot 4 \\ &= 1 \cdot 1 \cdot 2 \cdot 3 \cdot 4 \end{aligned}$$

Recursive Functions

- Example 4: Define a recursive function that computes n factorial: $n! = 1 \cdot 2 \cdot \ldots \cdot n$ where n is a natural number

$$
\begin{aligned}
f(4) &= f(3) \cdot 4 \\
&= f(2) \cdot 3 \cdot 4 \\
&= f(1) \cdot 2 \cdot 3 \cdot 4 \\
&= f(0) \cdot 1 \cdot 2 \cdot 3 \cdot 4 \\
&= 1 \cdot 1 \cdot 2 \cdot 3 \cdot 4 \\
&= 24
\end{aligned}
$$

The Fibonacci Sequence

- The Fibonacci sequence is

$$0, 1, 1, 2, 3, 5, 8, 13, \cdots$$

The Fibonacci Sequence

- The Fibonacci sequence is

$$0, 1, 1, 2, 3, 5, 8, 13, \cdots$$

The first two numbers of the sequence are 0 and 1. Each other number in the sequence is the sum of its two previous numbers in the sequence

The Fibonacci Sequence

- The Fibonacci function computes values in the Fibonacci sequence

$$f(0) = 0$$

$$f(1) = 1$$

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

The Fibonacci Sequence

- The Fibonacci function computes values in the Fibonacci sequence

$$f(4) \quad = f(2) + f(3)$$

The Fibonacci Sequence

- The Fibonacci function computes values in the Fibonacci sequence

$$
\begin{aligned}
f(4) &= f(2) + f(3) \\
&= f(0) + f(1) + f(3)
\end{aligned}
$$

The Fibonacci Sequence

- The Fibonacci function computes values in the Fibonacci sequence

$$
\begin{aligned}
f(4) &= f(2) + f(3) \\
&= f(0) + f(1) + f(3) \\
&= 0 + f(1) + f(3)
\end{aligned}
$$

The Fibonacci Sequence

- The Fibonacci function computes values in the Fibonacci sequence

$$
\begin{aligned}
f(4) &= f(2) + f(3) \\
&= f(0) + f(1) + f(3) \\
&= 0 + f(1) + f(3) \\
&= 0 + 1 + f(3)
\end{aligned}
$$

The Fibonacci Sequence

- The Fibonacci function computes values in the Fibonacci sequence

$$
\begin{aligned}
f(4) &= f(2) + f(3) \\
&= f(0) + f(1) + f(3) \\
&= 0 + f(1) + f(3) \\
&= 0 + 1 + f(3) \\
&= 0 + 1 + f(1) + f(2)
\end{aligned}
$$

The Fibonacci Sequence

- The Fibonacci function computes values in the Fibonacci sequence

$$
\begin{aligned}
f(4) &= f(2) + f(3) \\
&= f(0) + f(1) + f(3) \\
&= 0 + f(1) + f(3) \\
&= 0 + 1 + f(3) \\
&= 0 + 1 + f(1) + f(2) \\
&= 0 + 1 + 1 + f(2)
\end{aligned}
$$

The Fibonacci Sequence

- The Fibonacci function computes values in the Fibonacci sequence

$$
\begin{aligned}
f(4) &= f(2) + f(3) \\
&= f(0) + f(1) + f(3) \\
&= 0 + f(1) + f(3) \\
&= 0 + 1 + f(3) \\
&= 0 + 1 + f(1) + f(2) \\
&= 0 + 1 + 1 + f(2) \\
&= 0 + 1 + 1 + f(0) + f(1)
\end{aligned}
$$

The Fibonacci Sequence

- The Fibonacci function computes values in the Fibonacci sequence

$$
\begin{aligned}
f(4) \quad &= f(2) + f(3) \\
&= f(0) + f(1) + f(3) \\
&= 0 + f(1) + f(3) \\
&= 0 + 1 + f(3) \\
&= 0 + 1 + f(1) + f(2) \\
&= 0 + 1 + 1 + f(2) \\
&= 0 + 1 + 1 + f(0) + f(1) \\
&= 0 + 1 + 1 + 0 + f(1)
\end{aligned}
$$

The Fibonacci Sequence

- The Fibonacci function computes values in the Fibonacci sequence

$$
\begin{aligned}
f(4) &= f(2) + f(3) \\
&= f(0) + f(1) + f(3) \\
&= 0 + f(1) + f(3) \\
&= 0 + 1 + f(3) \\
&= 0 + 1 + f(1) + f(2) \\
&= 0 + 1 + 1 + f(2) \\
&= 0 + 1 + 1 + f(0) + f(1) \\
&= 0 + 1 + 1 + 0 + f(1) \\
&= 0 + 1 + 1 + 0 + 1
\end{aligned}
$$

The Fibonacci Sequence

- The Fibonacci function computes values in the Fibonacci sequence

$$
\begin{aligned}
f(4) &= f(2) + f(3) \\
&= f(0) + f(1) + f(3) \\
&= 0 + f(1) + f(3) \\
&= 0 + 1 + f(3) \\
&= 0 + 1 + f(1) + f(2) \\
&= 0 + 1 + 1 + f(2) \\
&= 0 + 1 + 1 + f(0) + f(1) \\
&= 0 + 1 + 1 + 0 + f(1) \\
&= 0 + 1 + 1 + 0 + 1 \\
&= 3
\end{aligned}
$$

Recursively Defined Sets

- Recursively defined sets use recursion to specify the elements in a set
  1. Base elements of the set are explicitly defined
  2. A recursive rule is given to define additional elements in the set

- Recursively defined sets are also known as inductively defined sets

Recursively Defined Sets

- Example: A recursive definition of $\boldsymbol{N}$, the set of natural numbers

1. $0 \in \boldsymbol{N}$
2. If $x \in \boldsymbol{N}$, then $x + 1 \in \boldsymbol{N}$
3. Nothing else is in $\boldsymbol{N}$

Recursively Defined Sets

- Example 5: A recursive definition of a subset S of natural numbers

1. $3 \in S$
2. If $x \in S$ and $y \in S$, then $x + y \in S$
3. Nothing else is in S

Recursively Defined Sets

- Example: A recursive definition of the set of properly nested parentheses, P

1. $( ) \in P$
2. If $u \in P$ and $v \in P$, then $(u) \in P$ and $uv \in P$
3. Nothing else is in P

Recursively Defined Set of Binary Strings

- The set of binary strings of any finite, non-negative length, B^* has a recursive definition

1. $\lambda \in B^*$ where λ is the empty string
2. If $s \in B^*$, then $s0 \in B^*$ and $s1 \in B^*$
3. Nothing else is in B^*

Recursively Defined Set of Strings

- Some members of B^*
  - $\lambda \in B^*$
  - $\lambda 0 = 0$, so $0 \in B^*$
  - $\lambda 1 = 1$, so $1 \in B^*$
  - $00 \in B^*$
  - $10 \in B^*$
  - $01 \in B^*$
  - $11 \in B^*$

Recursive Functions on Binary Strings

- Let $|\cdot|: B^* \rightarrow \boldsymbol{N}$ be a function that recursively computes the length of a string
  - Note that $|\cdot|$ is a function that is called by replacing the dot with an argument

$$|\lambda| = 0$$

$$|s0| = 1 + |s|$$

$$|s1| = 1 + |s|$$

Recursive Functions on Strings

$$|0110| = 1 + |011|$$

Recursive Functions on Strings

$$
\begin{aligned}
|0110| &= 1 + |011| \\
&= 1 + 1 + |01|
\end{aligned}
$$

Recursive Functions on Strings

$$
\begin{aligned}
|0110| &= 1 + |011| \\
&= 1 + 1 + |01| \\
&= 1 + 1 + 1 + |1|
\end{aligned}
$$

Recursive Functions on Strings

$$
\begin{aligned}
|0110| &= 1 + |011| \\
&= 1 + 1 + |01| \\
&= 1 + 1 + 1 + |0| \\
&= 1 + 1 + 1 + 1 + |\lambda|
\end{aligned}
$$

Recursive Functions on Strings

$$
\begin{aligned}
|0110| &= 1 + |011| \\
&= 1 + 1 + |01| \\
&= 1 + 1 + 1 + |1| \\
&= 1 + 1 + 1 + 1 + |\lambda| \\
&= 1 + 1 + 1 + 1 + 0
\end{aligned}
$$

Recursive Functions on Strings

$$
\begin{aligned}
|0110| &= 1 + |011| \\
&= 1 + 1 + |01| \\
&= 1 + 1 + 1 + |1| \\
&= 1 + 1 + 1 + 1 + |\lambda| \\
&= 1 + 1 + 1 + 1 + 0 \\
&= 4
\end{aligned}
$$

Recursive Functions on Strings

- Compare the recursive definition of B^* to the recursive definition of $|\cdot|: B^* \rightarrow \boldsymbol{N}$

$\lambda \in B^*$	$\lvert\lambda\rvert = 0$
$s0 \in B^*$ if $s \in B^*$	$\lvert s0\rvert = 1 + \lvert s\rvert$
$s1 \in B^*$ if $s \in B^*$	$\lvert s1\rvert = 1 + \lvert s\rvert$

Counting Digits

- Example: Define a recursive function that counts the number of digits in a natural number

- First attempt

$$length(0) = 1$$

$$length(n + 1) = ???$$

Counting Digits

- Consider a different definition of the natural numbers
  1. If $d \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$, then $d \in \boldsymbol{N}$
  2. If $n \in \boldsymbol{N}$ and $d \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$, then $10n + d \in \boldsymbol{N}$

Counting Digits

- Consider a different definition of the natural numbers
  1. If $d \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$, then $d \in \boldsymbol{N}$
  2. If $n \in \boldsymbol{N}$ and $d \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$, then $10n + d \in \boldsymbol{N}$

  Examples:

  $4 \in \boldsymbol{N}$

  $45 \in \boldsymbol{N}$

  $451 \in \boldsymbol{N}$

Counting Digits

- Consider a different definition of the natural numbers
  1. If $d \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$, then $d \in \boldsymbol{N}$
  2. If $n \in \boldsymbol{N}$ and $d \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$, then $10n + d \in \boldsymbol{N}$

$length(0) = 1 \quad length(1) = 1 \quad length(2) = 1 \quad length(3) = 1 \quad length(4) = 1$

$length(5) = 1 \quad length(6) = 1 \quad length(7) = 1 \quad length(8) = 1 \quad length(9) = 1$

$length(10n + d) = 1 + length(n)$

Counting Digits

$length(0) = 1 \quad length(1) = 1 \quad length(2) = 1 \quad length(3) = 1 \quad length(4) = 1$

$length(5) = 1 \quad length(6) = 1 \quad length(7) = 1 \quad length(8) = 1 \quad length(9) = 1$

$length(10n + d) = 1 + length(n)$

$length(451)$

$length(451)$

Counting Digits

$length(0) = 1 \quad length(1) = 1 \quad length(2) = 1 \quad length(3) = 1 \quad length(4) = 1$

$length(5) = 1 \quad length(6) = 1 \quad length(7) = 1 \quad length(8) = 1 \quad length(9) = 1$

$length(10n + d) = 1 + length(n)$

$length(451) = 1 + length(45)$

$length(451)$

Counting Digits

$length(0) = 1 \quad length(1) = 1 \quad length(2) = 1 \quad length(3) = 1 \quad length(4) = 1$

$length(5) = 1 \quad length(6) = 1 \quad length(7) = 1 \quad length(8) = 1 \quad length(9) = 1$

$length(10n + d) = 1 + length(n)$

$length(451) = 1 + length(45)$

$length(451) = 1 + 1 + length(4)$

Counting Digits

$length(0) = 1 \quad length(1) = 1 \quad length(2) = 1 \quad length(3) = 1 \quad length(4) = 1$

$length(5) = 1 \quad length(6) = 1 \quad length(7) = 1 \quad length(8) = 1 \quad length(9) = 1$

$length(10n + d) = 1 + length(n)$

$$\begin{aligned} length(451) &= 1 + length(45) \\ length(451) &= 1 + 1 + length(4) \\ &= 1 + 1 + 1 \end{aligned}$$

Counting Digits

$length(0) = 1 \quad length(1) = 1 \quad length(2) = 1 \quad length(3) = 1 \quad length(4) = 1$

$length(5) = 1 \quad length(6) = 1 \quad length(7) = 1 \quad length(8) = 1 \quad length(9) = 1$

$length(10n + d) = 1 + length(n)$

$$
\begin{aligned}
length(451) &= 1 + length(45) \\
length(451) &= 1 + 1 + length(4) \\
&= 1 + 1 + 1 \\
&= 3
\end{aligned}
$$

Counting Digits

$length(0) = 1 \quad length(1) = 1 \quad length(2) = 1 \quad length(3) = 1 \quad length(4) = 1$

$length(5) = 1 \quad length(6) = 1 \quad length(7) = 1 \quad length(8) = 1 \quad length(9) = 1$

$length(10n + d) = 1 + length(n)$

Note that $n = \left\lfloor \frac{(10n+d)}{10} \right\rfloor$

Counting Digits

- Rewrite as a function in pseudocode
  - Name: length
  - Input: a natural number n
  - Output: The number of digits in n

```
if n <= 9
  return 1
else
  return 1 + length(floor(n/10))
end-if
```