Section 8.10
Recursive Algorithms

Recursive Algorithms

- An algorithm is recursive if it solves a problem by reducing it to an instance of the same problem with a simpler input

Factorial

- Example 1: Create a recursive algorithm for computing $n!$ where n is a non-negative integer
  - Write out $0!, 1!, 2!, 3!, \dots$

$$1, \quad 1 \cdot 1, \quad 1 \cdot 1 \cdot 2, \quad 1 \cdot 1 \cdot 2 \cdot 3$$

The nth term is n times the previous term

Factorial

- Example 1: Create a recursive algorithm for computing $n!$ where n is a non-negative integer
  - Define the function recursively

$$f(0) = 1$$

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

Factorial

- Example 1: Create a recursive algorithm for computing $n!$ where n is a non-negative integer
  - Test it

$$\begin{aligned} f(5) &= 5 \cdot f(4) \\ &= 5 \cdot 4 \cdot f(3) \\ &= 5 \cdot 4 \cdot 3 \cdot f(2) \\ &= 5 \cdot 4 \cdot 3 \cdot 2 \cdot f(1) \\ &= 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1 \cdot f(0) \\ &= 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1 \cdot 1 \\ &= 120 \end{aligned}$$

Factorial

- Example 1: Create a recursive algorithm for computing $n!$ where n is a non-negative integer
    - Code it

Name: factorial

Input: non-negative integer n

Output: n!

```
if (n = 0)
  return 1
else
  return n * factorial(n-1)
end-if
```

Exponentiation

- Example 2: Create a recursive algorithm for computing a^n where n is a non-negative integer
  - Write out $a^0, a^1, a^2, a^3, \dots$

$$1, \;\; 1 \cdot a, \;\; 1 \cdot a \cdot a, \;\; 1 \cdot a \cdot a \cdot a$$

The nth term is a times the previous term

Exponentiation

- Example 2: Create a recursive algorithm for computing a^n where n is a non-negative integer
  - Define the function recursively

$$f(a, 0) = 1$$

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

Exponentiation

- Example 2: Create a recursive algorithm for computing a^n where n is a non-negative integer
  - Test it

$$\begin{aligned} f(3,5) &= 3 \cdot f(3,4) \\ &= 3 \cdot 3 \cdot f(3,3) \\ &= 3 \cdot 3 \cdot 3 \cdot f(3,2) \\ &= 3 \cdot 3 \cdot 3 \cdot 3 \cdot f(3,1) \\ &= 3 \cdot 3 \cdot 3 \cdot 3 \cdot 3 \cdot f(3,0) \\ &= 3 \cdot 3 \cdot 3 \cdot 3 \cdot 3 \cdot 1 \\ &= 243 \end{aligned}$$

Exponentiation

- Example 2: Create a recursive algorithm for computing a^n where n is a non-negative integer
    - Code it

Name: power

Input: real number a and nonnegative integer n

Output: a^n

```
if n = 0
  return 1
else
  return a * power(a, n-1)
end-if
```

Summation

- Example 3: Create a recursive algorithm for computing $\sum_{i=0}^{n} i$ where n is a non-negative integer
  - Write out $\sum_{i=0}^{0} i$, $\sum_{i=0}^{1} i$, $\sum_{i=0}^{2} i$, $\sum_{i=0}^{3} i$, …

$$0, \quad 0+1, \quad 0+1+2, \quad 0+1+2+3$$

The nth term is n plus the previous term

Summation

- Example 3: Create a recursive algorithm for computing $\sum_{i=0}^{n} i$ where n is a non-negative integer
  - Define the function recursively

$$f(0) = 0$$

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

Summation

- Example 3: Create a recursive algorithm for computing $\sum_{i=0}^{n} i$ where n is a non-negative integer
  - Test it

$$
\begin{aligned}
f(5) &= 5 + f(4) \\
&= 5 + 4 + f(3) \\
&= 5 + 4 + 3 + f(2) \\
&= 5 + 4 + 3 + 2 + f(1) \\
&= 5 + 4 + 3 + 2 + 1 + f(0) \\
&= 5 + 4 + 3 + 2 + 1 + 0 \\
&= 15
\end{aligned}
$$

Summation

- Example 3: Create a recursive algorithm for computing $\sum_{i=0}^{n} i$ where n is a non-negative integer
  - Code it

```
summation
Input: nonnegative integer n
Output: $\sum_{i=0}^{n} i$

if n = 0
  return 0
else
  return n + summation(n-1)
end-if
```

Fibonacci Numbers

- Example 4: Create a recursive algorithm for computing the nth Fibonacci number where n is a non-negative integer
  - Write out the Fibonacci numbers

$$0, \quad 1, \quad 0+1, \quad 1+1, \quad 1+2$$

The nth term is the sum of the two previous terms

Fibonacci Numbers

- Example 4: Create a recursive algorithm for computing the nth Fibonacci number where n is a non-negative integer
  - Define the function recursively

$$f(0) = 0$$

$$f(1) = 1$$

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

Fibonacci Numbers

- Example 4: Create a recursive algorithm for computing the nth Fibonacci number where n is a non-negative integer
  - Test it

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

Fibonacci Numbers

- Example 4: Create a recursive algorithm for computing the nth Fibonacci number where n is a non-negative integer
  - Code it

fibonacci
Input: nonnegative integer n
Output: the nth Fibonacci number

```
if n = 0 return 0 end-if
if n = 1 return 1 end-if
return fibonacci(n-2) + fibonacci(n-1)
```

Fibonacci Numbers

- Example 4: Create a recursive algorithm for computing the nth Fibonacci number where n is a non-negative integer
    - How does the algorithm perform when computing `fibonacci(5)`?
    - How can the Fibonacci numbers be computed using a loop instead of recursion?

Summing a Sequence

- Example: Summing a sequence of numbers

Name: summation
Input: a_1, a_2, a_3, …, a_n
Output: $a_1 + a_2 + a_3 + … + a_n$

```
sum := 0
for i:= 1 to n
  sum := sum + $a_i$
end-for
return sum
```

Summing a Sequence

- Example: Summing a sequence of numbers

Name: summation
Input: $a_1, a_2, a_3, \ldots, a_n$
Output: $a_1 + a_2 + a_3 + \ldots + a_n$

```
sum := 0
for i:= 1 to n
  sum := sum + $a_i$
end-for
return sum
```

a	7	3	1	5
	1	2	3	4

sum	i

Summing a Sequence

- Example: Summing a sequence of numbers

Name: summation
Input: a_1, a_2, a_3, …, a_n
Output: $a_1 + a_2 + a_3 + … + a_n$

```
sum := 0
for i:= 1 to n
  sum := sum + $a_i$
end-for
return sum
```

a	7	3	1	5
	1	2	3	4

sum	i
0	

Summing a Sequence

- Example: Summing a sequence of numbers

Name: summation
Input: $a_1, a_2, a_3, \ldots, a_n$
Output: $a_1 + a_2 + a_3 + \ldots + a_n$

```
sum := 0
for i:= 1 to n
  sum := sum + $a_i$
end-for
return sum
```

a	7	3	1	5
	1	2	3	4

sum	i
0	1

Summing a Sequence

- Example: Summing a sequence of numbers

Name: summation
Input: $a_1, a_2, a_3, \ldots, a_n$
Output: $a_1 + a_2 + a_3 + \ldots + a_n$

```
sum := 0
for i:= 1 to n
  sum := sum + $a_i$
end-for
return sum
```

a	7	3	1	5
	1	2	3	4

sum	i
~~0~~	1
7	

Summing a Sequence

- Example: Summing a sequence of numbers

Name: summation
Input: $a_1, a_2, a_3, \ldots, a_n$
Output: $a_1 + a_2 + a_3 + \ldots + a_n$

```
sum := 0
for i:= 1 to n
  sum := sum + $a_i$
end-for
return sum
```

a	7	3	1	5
	1	2	3	4

sum	i
~~0~~	~~1~~
7	2

Summing a Sequence

- Example: Summing a sequence of numbers

Name: summation
Input: $a_1, a_2, a_3, \ldots, a_n$
Output: $a_1 + a_2 + a_3 + \ldots + a_n$

```
sum := 0
for i:= 1 to n
  sum := sum + $a_i$
end-for
return sum
```

a	7	3	1	5
	1	2	3	4

sum	i
~~0~~	~~1~~
~~7~~	2
10	

Summing a Sequence

- Example: Summing a sequence of numbers

Name: summation
Input: $a_1, a_2, a_3, \ldots, a_n$
Output: $a_1 + a_2 + a_3 + \ldots + a_n$

```
sum := 0
for i:= 1 to n
  sum := sum + $a_i$
end-for
return sum
```

a	7	3	1	5
	1	2	3	4

sum	i
~~0~~	~~1~~
~~7~~	~~2~~
10	3

Summing a Sequence

- Example: Summing a sequence of numbers

Name: summation
Input: $a_1, a_2, a_3, \ldots, a_n$
Output: $a_1 + a_2 + a_3 + \ldots + a_n$

```
sum := 0
for i:= 1 to n
  sum := sum + $a_i$
end-for
return sum
```

a	7	3	1	5
	1	2	3	4

sum	i
~~0~~	~~1~~
~~7~~	~~2~~
~~10~~	3
11	

Summing a Sequence

- Example: Summing a sequence of numbers

Name: summation
Input: $a_1, a_2, a_3, \ldots, a_n$
Output: $a_1 + a_2 + a_3 + \ldots + a_n$

```
sum := 0
for i:= 1 to n
  sum := sum + $a_i$
end-for
return sum
```

a	7	3	1	5
	1	2	3	4

sum	i
~~0~~	~~1~~
~~7~~	~~2~~
~~10~~	~~3~~
11	4

Summing a Sequence

- Example: Summing a sequence of numbers

Name: summation
Input: $a_1, a_2, a_3, \ldots, a_n$
Output: $a_1 + a_2 + a_3 + \ldots + a_n$

```
sum := 0
for i:= 1 to n
  sum := sum + $a_i$
end-for
return sum
```

a	7	3	1	5
	1	2	3	4

sum	i
~~0~~	~~1~~
~~7~~	~~2~~
~~10~~	~~3~~
~~11~~	4
16	

Summing a Sequence

- Example: Summing a sequence of numbers

Name: summation
Input: $a_1, a_2, a_3, \ldots, a_n$
Output: $a_1 + a_2 + a_3 + \ldots + a_n$

```
sum := 0
for i:= 1 to n
  sum := sum + a_i
end-for
return sum
```

a	7	3	1	5
	1	2	3	4

sum	i
~~0~~	~~1~~
~~7~~	~~2~~
~~10~~	~~3~~
~~11~~	4
16	

`sum = 5 + (1 + (3 + (7 + 0)))`

Summing a Sequence

- Example: Summing a sequence of numbers

Name: summation
Input: $a_1, a_2, a_3, \ldots, a_n$
Output: $a_1 + a_2 + a_3 + \ldots + a_n$

```
sum := 0
for i:= 1 to n
  sum := sum + $a_i$
end-for
return sum
```

a	7	3	1	5
	1	2	3	4

sum	i
~~0~~	~~1~~
~~7~~	~~2~~
~~10~~	~~3~~
~~11~~	4
16	

$$\text{sum} = a_4 + (a_3 + (a_2 + (a_1 + 0)))$$

Summing a Sequence

- Recursively summing a sequence of numbers

Name: recsum
Input: a_1, a_2, …, a_n
i
Output: $a_1 + a_2 + … + a_i$

```
if i = 0
  return 0
else
  return $a_i$ + recsum($a_1$, $a_2$, $a_3$, …, $a_n$, i-1)
end-if
```

Summing a Sequence

- Recursively summing a sequence of numbers

Name: recsum
Input: a_1, a_2, …, a_n
i
Output: $a_1 + a_2 + \ldots + a_i$

```
if i = 0
  return 0
else
  return $a_i$ + recsum($a_1$, $a_2$, $a_3$, …, $a_n$, i-1)
end-if
```

```
recsum(7, 3, 1, 5, 4)
```

Summing a Sequence

- Recursively summing a sequence of numbers

Name: recsum
Input: a_1, a_2, …, a_n
i
Output: $a_1 + a_2 + … + a_i$

```
if i = 0
  return 0
else
  return $a_i$ + recsum($a_1$, $a_2$, $a_3$, …, $a_n$, i-1)
end-if
```

```
recsum(7, 3, 1, 5, 4)
5 + recsum(7, 3, 1, 5, 3)
```

Summing a Sequence

- Recursively summing a sequence of numbers

Name: recsum
Input: $a_1, a_2, \ldots, a_n$
i
Output: $a_1 + a_2 + \ldots + a_i$

```
if i = 0
  return 0
else
  return $a_i$ + recsum($a_1$, $a_2$, $a_3$, …, $a_n$, i-1)
end-if
```

```
recsum(7, 3, 1, 5, 4)
5 + recsum(7, 3, 1, 5, 3)
5 + 1 + recsum(7, 3, 1, 5, 2)
```

Summing a Sequence

- Recursively summing a sequence of numbers

Name: recsum
Input: $a_1, a_2, \ldots, a_n$
i
Output: $a_1 + a_2 + \ldots + a_i$

```
if i = 0
  return 0
else
  return $a_i$ + recsum($a_1$, $a_2$, $a_3$, …, $a_n$, i-1)
end-if
```

```
recsum(7, 3, 1, 5, 4)
5 + recsum(7, 3, 1, 5, 3)
5 + 1 + recsum(7, 3, 1, 5, 2)
5 + 1 + 3 + recsum(7, 3, 1, 5, 1)
```

Summing a Sequence

- Recursively summing a sequence of numbers

Name: recsum
Input: $a_1, a_2, \ldots, a_n$
i
Output: $a_1 + a_2 + \ldots + a_i$

```
if i = 0
  return 0
else
  return $a_i$ + recsum($a_1$, $a_2$, $a_3$, …, $a_n$, i-1)
end-if
```

```
recsum(7, 3, 1, 5, 4)
5 + recsum(7, 3, 1, 5, 3)
5 + 1 + recsum(7, 3, 1, 5, 2)
5 + 1 + 3 + recsum(7, 3, 1, 5, 1)
5 + 1 + 3 + 7 + recsum(7, 3, 1, 5, 0)
```

Summing a Sequence

- Recursively summing a sequence of numbers

Name: recsum
Input: $a_1, a_2, \ldots, a_n$
i
Output: $a_1 + a_2 + \ldots + a_i$

```
if i = 0
  return 0
else
  return $a_i$ + recsum($a_1$, $a_2$, $a_3$, …, $a_n$, i-1)
end-if
```

```
recsum(7, 3, 1, 5, 4)
5 + recsum(7, 3, 1, 5, 3)
5 + 1 + recsum(7, 3, 1, 5, 2)
5 + 1 + 3 + recsum(7, 3, 1, 5, 1)
5 + 1 + 3 + 7 + recsum(7, 3, 1, 5, 0)
5 + 1 + 3 + 7 + 0
```

Summing a Sequence

- Recursively summing a sequence of numbers

Name: recsum
Input: $a_1, a_2, \ldots, a_n$
i
Output: $a_1 + a_2 + \ldots + a_i$

```
if i = 0
  return 0
else
  return $a_i$ + recsum($a_1$, $a_2$, $a_3$, …, $a_n$, i-1)
end-if
```

```
recsum(7, 3, 1, 5, 4)
5 + recsum(7, 3, 1, 5, 3)
5 + 1 + recsum(7, 3, 1, 5, 2)
5 + 1 + 3 + recsum(7, 3, 1, 5, 1)
5 + 1 + 3 + 7 + recsum(7, 3, 1, 5, 0)
5 + 1 + 3 + 7 + 0
16
```

Linear Search

- Another example: Finding the first occurrence of a number in a of a sequence of numbers

Name: linearsearch

Input: x, i, a_1, a_2, a_3, …, a_n

Output: The index of the first occurrence of x in a_1, a_2, a_3, …, a_n starting from i or 0 if x is not in the sequence

```
if i > n
  return 0
else
  if x = $a_i$
    return i
  else
    return linearsearch(x, i+1, $a_1$, $a_2$, $a_3$, …, $a_n$)
  end-if
end-if
```

Linear Search

- Another example: Finding the first occurrence of a number in a of a sequence of numbers

  Name: linearsearch

  Input: x, i, a_1, a_2, a_3, …, a_n

  Output: The index of the first occurrence of x in a_1, a_2, a_3, …, a_n starting from i or 0 if x is not in the sequence

```
if i > n
  return 0
else
  if x = $a_i$
    return i
  else
    return linearsearch(x, i+1, $a_1$, $a_2$, $a_3$, …, $a_n$)
  end-if
end-if
```

```
linearsearch(3, 1, 7, 3, 1, 5)
```

Linear Search

- Another example: Finding the first occurrence of a number in a of a sequence of numbers

Name: linearsearch

Input: x, i, a_1, a_2, a_3, …, a_n

Output: The index of the first occurrence of x in a_1, a_2, a_3, …, a_n starting from i or 0 if x is not in the sequence

```
if i > n
  return 0
else
  if x = $a_i$
    return i
  else
    return linearsearch(x, i+1, $a_1$, $a_2$, $a_3$, …, $a_n$)
  end-if
end-if
```

```
linearsearch(3, 1, 7, 3, 1, 5)
linearsearch(3, 2, 7, 3, 1, 5)
```

Linear Search

- Another example: Finding the first occurrence of a number in a of a sequence of numbers

Name: linearsearch

Input: x, i, a_1, a_2, a_3, …, a_n

Output: The index of the first occurrence of x in a_1, a_2, a_3, …, a_n starting from i or 0 if x is not in the sequence

```
if i > n
  return 0
else
  if x = $a_i$
    return i
  else
    return linearsearch(x, i+1, $a_1$, $a_2$, $a_3$, …, $a_n$)
  end-if
end-if
```

```
linearsearch(3, 1, 7, 3, 1, 5)
linearsearch(3, 2, 7, 3, 1, 5)
2
```

Linear Search

- Another example: Finding the first occurrence of a number in a of a sequence of numbers

Name: linearsearch

Input: x, i, a_1, a_2, a_3, …, a_n

Output: The index of the first occurrence of x in a_1, a_2, a_3, …, a_n starting from i or 0 if x is not in the sequence

```
if i > n
  return 0
else
  if x = $a_i$
    return i
  else
    return linearsearch(x, i+1, $a_1$, $a_2$, $a_3$, …, $a_n$)
  end-if
end-if
```

```
linearsearch(9, 1, 7, 3, 1, 5)
```

Linear Search

- Another example: Finding the first occurrence of a number in a of a sequence of numbers

  Name: linearsearch

  Input: x, i, a_1, a_2, a_3, …, a_n

  Output: The index of the first occurrence of x in a_1, a_2, a_3, …, a_n starting from i or 0 if x is not in the sequence

```
if i > n
  return 0
else
  if x = $a_i$
    return i
  else
    return linearsearch(x, i+1, $a_1$, $a_2$, $a_3$, …, $a_n$)
  end-if
end-if
```

```
linearsearch(9, 1, 7, 3, 1, 5)
linearsearch(9, 2, 7, 3, 1, 5)
```

Linear Search

- Another example: Finding the first occurrence of a number in a of a sequence of numbers

  Name: linearsearch

  Input: x, i, $a_1, a_2, a_3, \dots, a_n$

  Output: The index of the first occurrence of x in $a_1, a_2, a_3, \dots, a_n$ starting from i or 0 if x is not in the sequence

```
if i > n
  return 0
else
  if x = $a_i$
    return i
  else
    return linearsearch(x, i+1, $a_1$, $a_2$, $a_3$, …, $a_n$)
  end-if
end-if
```

```
linearsearch(9, 1, 7, 3, 1, 5)
linearsearch(9, 2, 7, 3, 1, 5)
linearsearch(9, 3, 7, 3, 1, 5)
```

Linear Search

- Another example: Finding the first occurrence of a number in a of a sequence of numbers

Name: linearsearch

Input: x, i, a_1, a_2, a_3, …, a_n

Output: The index of the first occurrence of x in a_1, a_2, a_3, …, a_n starting from i or 0 if x is not in the sequence

```
if i > n
  return 0
else
  if x = $a_i$
    return i
  else
    return linearsearch(x, i+1, $a_1$, $a_2$, $a_3$, …, $a_n$)
  end-if
end-if
```

```
linearsearch(9, 1, 7, 3, 1, 5)
linearsearch(9, 2, 7, 3, 1, 5)
linearsearch(9, 3, 7, 3, 1, 5)
linearsearch(9, 4, 7, 3, 1, 5)
```

Linear Search

- Another example: Finding the first occurrence of a number in a of a sequence of numbers

Name: linearsearch

Input: x, i, a_1, a_2, a_3, …, a_n

Output: The index of the first occurrence of x in a_1, a_2, a_3, …, a_n starting from i or 0 if x is not in the sequence

```
if i > n
  return 0
else
  if x = $a_i$
    return i
  else
    return linearsearch(x, i+1, $a_1$, $a_2$, $a_3$, …, $a_n$)
  end-if
end-if
```

```
linearsearch(9, 1, 7, 3, 1, 5)
linearsearch(9, 2, 7, 3, 1, 5)
linearsearch(9, 3, 7, 3, 1, 5)
linearsearch(9, 4, 7, 3, 1, 5)
linearsearch(9, 5, 7, 3, 1, 5)
```

Linear Search

- Another example: Finding the first occurrence of a number in a of a sequence of numbers

  Name: linearsearch

  Input: x, i, a_1, a_2, a_3, …, a_n

  Output: The index of the first occurrence of x in a_1, a_2, a_3, …, a_n starting from i or 0 if x is not in the sequence

```
if i > n
  return 0
else
  if x = $a_i$
    return i
  else
    return linearsearch(x, i+1, $a_1$, $a_2$, $a_3$, …, $a_n$)
  end-if
end-if
```

```
linearsearch(9, 1, 7, 3, 1, 5)
linearsearch(9, 2, 7, 3, 1, 5)
linearsearch(9, 3, 7, 3, 1, 5)
linearsearch(9, 4, 7, 3, 1, 5)
linearsearch(9, 5, 7, 3, 1, 5)
0
```