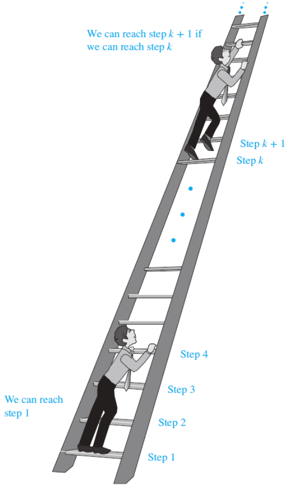
Chapter 4
Induction and Recursion

MAD101

Ly Anh Duong

duongla3@fe.edu.vn

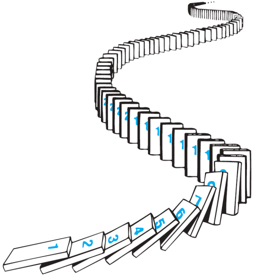
Table of Contents



We can reach step $k + 1$ if we can reach step k
Step $k + 1$
Step k
Step 4
Step 3
Step 2
Step 1
We can reach step 1

To prove that $P(n)$ is true for all positive integers n, where $P(n)$ is a propositional function, we complete two steps:

- **BASIS STEP.** We verify that $P(1)$ is true.
- **INDUCTIVE STEP.** We show that the conditional statement $P(k) \to P(k+1)$ is true for all positive integers k.



Examples.

1. Prove that $1 + 2 + 3 + ... + n - 1 + n = \dfrac{n(n+1)}{2}$
2. Conjecture a formula for the sum of the first n positive odd integers. Then prove your conjecture using mathematical induction.
3. $2^0 + 2^1 + ... + 2^n = 2^{n+1} - 1$

Solution

1 Mathematical Induction

Let P(n): " $1+2+3+\ldots+n = n(n+1)/2$".

- Basis step: P(1) : "$1 = 1(1+1)/2$" ➔ true
- Inductive step: With arbitrary $k>0$,

  P(k) :" $1+2+\ldots+k = k(k+1)/2$" is true.

We have

$$\underline{1+2+3+\ldots+k}+(k+1) = k(k+1)/2 + (k+1)$$
$$= [k(k+1) + 2(k+1)]/2$$
$$= (k+1)(k+2)/2$$
$$= (k+1)((k+1)+1)/2$$

P(k+1):"$1+2+3+\ldots+(k+1) = (k+1)(k+2)/2$" is true.

P(k) → P(k+1): true

Proved.

Solution

1 Mathematical Induction

The sum of the first n positive odd integers for n=1, 2, 3, 4, 5 are:

1=1, 1+3=4, 1+3+5=9,

1+3+5+7=16, 1+3+5+7+9=25.

- *Conjecture*: **$1+3+5+\ldots+(2n-1)=n^2$.**
- *Proof*: Let $P(n)$="**$1+3+5+\ldots+(2n-1)=n^2$.**"

- Basis step. $P(1)$="1=1" is true.

- Inductive step. $(P(k) \rightarrow P(k+1))$ is true.

Suppose $P(k)$ is true for arbitrary $k > 0$. That is, "$1+3+5+\ldots+(2k-1)=k^2$"

We have, $\underline{1+3+5+\ldots+(2k-1)}+(2k+1)=\underline{k^2}+2k+1 = (k+1)^2$.

So, $P(k+1)$ is true.

Proved.

Solution

1 Mathematical Induction

Solution: Let $P(n)$ be the proposition that $1 + 2 + 2^2 + \cdots + 2^n = 2^{n+1} - 1$ for the integer n.

BASIS STEP: $P(0)$ is true because $2^0 = 1 = 2^1 - 1$. This completes the basis step.

INDUCTIVE STEP: For the inductive hypothesis, we assume that $P(k)$ is true for an arbitrary nonnegative integer k. That is, we assume that

$$1 + 2 + 2^2 + \cdots + 2^k = 2^{k+1} - 1.$$

To carry out the inductive step using this assumption, we must show that when we assume that $P(k)$ is true, then $P(k+1)$ is also true. That is, we must show that

$$1 + 2 + 2^2 + \cdots + 2^k + 2^{k+1} = 2^{(k+1)+1} - 1 = 2^{k+2} - 1$$

assuming the inductive hypothesis $P(k)$. Under the assumption of $P(k)$, we see that

$$\begin{aligned} 1 + 2 + 2^2 + \cdots + 2^k + 2^{k+1} &= (1 + 2 + 2^2 + \cdots + 2^k) + 2^{k+1} \\ &\overset{\text{IH}}{=} (2^{k+1} - 1) + 2^{k+1} \\ &= 2 \cdot 2^{k+1} - 1 \\ &= 2^{k+2} - 1. \end{aligned}$$

Note that we used the inductive hypothesis in the second equation in this string of equalities to replace $1 + 2 + 2^2 + \cdots + 2^k$ by $2^{k+1} - 1$. We have completed the inductive step.

Because we have completed the basis step and the inductive step, by mathematical induction we know that $P(n)$ is true for all nonnegative integers n. That is, $1 + 2 + \cdots + 2^n = 2^{n+1} - 1$ for all nonnegative integers n.

Strong Induction

To prove P(n) is true for all positive integers n, where P(n) is a propositional function, two steps are performed:

- **Basis step:** Verifying $P(1)$ is true
- **Inductive step:** Show $[P(1) \wedge P(2) \wedge \ldots \wedge P(k)] \to P(k+1)$ is true for all $k > 0$

Example 1. Prove that if n is an integer greater than 1, then n can be written as the product of primes.
Example 2. Prove that every amount of postage of 12 cents or more can be formed using just 4-cents and 5-cents stamps.

Solution. Example 1:

Let P(n): "n can be written as the product of primes".

- **Basis step:**

  $P(2) = 2$ OK
  $P(3) = 3$ OK
  $P(4) = 4 = 2.2$ OK
  ...

- **Inductive step:**

  Suppose $P(j)$ OK $\forall j \leq k$.
  Show $P(j)$ is true with $j = k + 1$:

  a. If $k + 1$ is a prime, then $P(k + 1)$ is true.
  b. If $k + 1$ is a composite, then $k + 1 = ab, 2 \leq a \leq b < k + 1$. Because $a, b < k + 1$, according to hypothesis, a and b can be written as a product of primes. Hence, $k + 1$ can be written as a product of primes.

Solution. Example 2

Let P(n): "n cents can be formed using just 4-cent and 5-cent stamps".

- **Basis step:**

  P(12) is true : $12cents = 3.4cents$ OK
  P(13) is true : $13 = 2.4 + 1.5$ OK
  P(14) is true : $14 = 1.4 + 2.5$ OK
  P(15) is true : $15 = 3.5$
  ...

- **Inductive step:**

  Suppose $P(j)\, is\, true with 12 j k, k > 15 \rightarrow P(k-3)\, is\, true.$
  Show $P(j)$ is true with $j = k+1$: We have: $k+1 = (k-3)+4, k > 12$

$\Longrightarrow$ P(k+1) is true because k+1 is the result of adding a 4-cent stamp to the amount k-3.
(Proved).

Table of Contents

2 Recursive

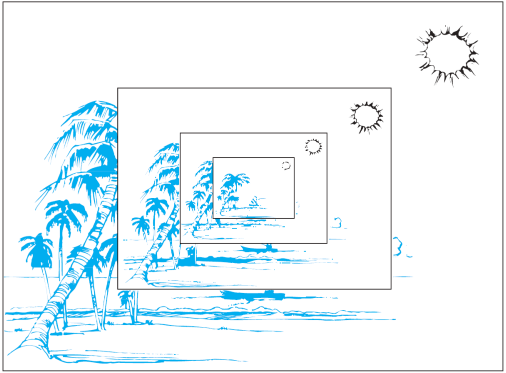
Fibonacci numbers:

$$1, 1, 2, 3, 5, 8, 13, 21, 34, 55, \ldots$$

n	1	2	3	4	5	6	7
F_n	1	1	2	3	5	8	13

1. Find F_8.
2. Find F_{20} if $F_{18} = 2584, F_{19} = 4181$.

Ingeneral, $F_1 = F_2 = 1, F_n = F_{n-1} + F_{n-2}, n \geq 3$



Sometimes, it is difficult to define an object explicitly. However, it may be easy to define this object in terms of itself. This process is called **recursion**.

Recursive definition of Fibonacci numbers

Procedure Fibo(n:positive integer)
if $n = 1$ **or** $n = 2$
 return 1
else
 return Fibo($n-1$)+Fibo($n-2$)
What is output value if input $n = 5$?

- Basis step $F_1 = 1, F_2 = 1$
- recursive step $F_n = F_{n-1} + F_{n-2}, n \geq 3$.

Recursively Defined Functions (hàm đệ quy)

We use two steps to define a function with the set of nonnegative integers as its domain:

1. **BASIS STEP:** Specify the value of the function at zero.
2. **RECURSIVE STEP:** Give a rule for finding its value at an integer from its values at smaller assessment integers.

Give an algorithm to find pseudo-random numbers if
$x_0 = 1, x_{n+1} = (3x_n + 17) \bmod 22$
Procedure pseudo(n:positive integer)
if $n = 0$
 return 1
else
 return (3*pseudo($n-1$)+17) mod 22

Example.

Procedure sum(n:$n \geq 1$, integer)
if $n = 1$
 return 1
else
 return sum($n-1$)+n
If input $n = 4$, what is the value of output?
$1 + 2 + 3 + 4 = 10$

Example.

2 Recursive

Procedure sum(n:$n \geq 1$, integer)
if $n = 1$
 return 5
else
 return sum($n - 1$)
If input $n = 4$, what is the value of output?
5

Example.
2 Recursive

Find the recursive algorithm of $f(n) = 5n + 1, n \geq 1$

- Basis step: $f(1) = 6$
- Recursive step: $f(n) = f(n-1) + 5$

Hence, the algorithm:

Procedure f(n:$n \geq 1$, integer)

if $n = 1$

return 6

else

return f($n-1$)+5

Exercises

2 Recursive

1. $a_n = 5n - 2, n = 1, 2, 3...?$
2. $a_n = n, n = 1, 2, 3...$
3. $f(n) = 1 + 2 + 3 + ... + n, n = 1, 2, 3, ...$
4. $f(n) = 2022, \forall n$
5. .

```
procedure thuattoan(a: real number, n: non-negative integer)
if n = 0
    return 1
else
    return a*thuattoan(a,n-1)
```

What is the result after calling *thuattoan*(2,3)?

1. Give the recursive definition of a^n (where a is a nonzero real number and n is a nonnegative integer).
2. Suppose that f is defined recursively by $f(0) = 3, f(n+1) = 2f(n) + 3$. Find $f(1), f(2), f(3)$ and $f(4)$.
3. Give a recursive definition of $\sum\limits_{k=0}^{n} a_k$

Solutions

2 Recursive

1. $a^0 = 1, a^n = a.a^{n-1}, n \geq 1$. Let $f(n) = a^n$. Thus $f(0) = 1, f(n) = af(n-1), n \geq 1$
2. .

$$f(1) = 2f(0) + 3 = 2 \cdot 3 + 3 = 9,$$
$$f(2) = 2f(1) + 3 = 2 \cdot 9 + 3 = 21,$$
$$f(3) = 2f(2) + 3 = 2 \cdot 21 + 3 = 45,$$
$$f(4) = 2f(3) + 3 = 2 \cdot 45 + 3 = 93.$$

3. .

Solution: The first part of the recursive definition is

$$\sum_{k=0}^{0} a_k = a_0.$$

The second part is

$$\sum_{k=0}^{n+1} a_k = \left(\sum_{k=0}^{n} a_k \right) + a_{n+1}.$$

Recursively Defined Sets (tập đệ quy)

Recursive definitions of sets have two parts

- **Basis step.** An initial collection of elements is specified.
- **Recursive step.** Rules for forming new elements in the set from those already known to be in the set are provided.

Examples. Consider the subset S of the set of integers recursively defined by

- **Basis step.** $3 \in S$
- **Recursive step.** If $x \in S$ and $y \in S$, then $x + y \in S$.

$\rightarrow S = \{3, 6, 9, 12, 15, 18, 21, ...\}$

Examples.

Find S if

a. 1 is in S, if x is in S then x + 1 and x + 2 are in S $S = \{1, 2, 3, ...\}$
b. 1 is in S, if x is in S then x + 3 is in S $S = \{1, 4, 7, 10, ...\}$
c. 1, 2 are in S, if x is in S then x + 3 is in S $S = \{1, 2, 4, 5, 7, 8, ...\}$

Examples.

Give a recursive definition of each of these sets.

a. $A = \{2, 5, 8, 11, 14, ...\}$ $(2 \in A, x \in A \to x + 3 \in A)$

b. $B = \{..., -5, -1, 3, 7, 10, ...\}$ $(3 \in A, x \in A \to x + 4 \in B \wedge x - 4 \in B)$

c. $C = \{3, 12, 48, 192, 768, ...\}$ $(3 \in C, x \in C \to 4x \in C)$

The set $\sum^*$ of strings

2 Recursive

The set $\sum^*$ of strings over the fintite set (alphabet) $\sum$ is defined recursively by

- **BASIS STEP:** $\lambda \in \sum^*$ (where λ is the empty string containing no symbols).
- **RECURSIVE STEP:** If $w \in \sum^*$ and $x \in \sum$, then $wx \in \sum^*$.

Examples.

1. $\sum = \{1\}$
    - $\sum_0^* = \{\lambda\}$
    - $\sum_1^* = \{\lambda, 1\}$
    - $\sum_2^* = \{\lambda, 1, 11\}$
    - $\sum_3^* = \{\lambda, 1, 11, 111\}$
    - $\sum_4^* = \{\lambda, 1, 11, 111, 1111\}$
    - ....

Examples.

2 Recursive

2. $\sum = \{0, 1\}$
   - $\sum_0^* = \{\lambda\}$
   - $\sum_1^* = \{\lambda, 0, 1\}$
   - $\sum_2^* = \{\lambda, 0, 1, 00, 01, 10, 11\}$
   - $\sum_3^* = \{\lambda, 0, 1, 00, 01, 10, 11, 000, 001, 010, 011, 100, 101, 110, 111\}$
   - .....

Strings Concatenation (nối các xâu)

Two strings can be combined via the operation of *concatenation*. Let Σ be a set of symbols and Σ^* the set of strings formed from symbols in Σ. We can define the concatenation of two strings, denoted by $\cdot$, recursively as follows.

BASIS STEP: If $w \in \Sigma^*$, then $w \cdot \lambda = w$, where λ is the empty string.

RECURSIVE STEP: If $w_1 \in \Sigma^*$ and $w_2 \in \Sigma^*$ and $x \in \Sigma$, then $w_1 \cdot (w_2 x) = (w_1 \cdot w_2)x$.

Example. The concatenation of $w_1 = abra$ and $w_2 = cadabra$ is $w_1 w_2 = abracadabra$.

Length of a String

2 Recursive

The length of a string can be recursively defined by

- $l(\lambda) = 0$
- $l(wx) = l(w) + 1$ if $w \in \sum^*$ and $x \in \sum$.

Table of Contents

Definition

An algorithm is called **recursive** if it solves a problem by reducing it to an instance of the same problem with smaller input.

Examples. Find 4!

$$0! = 1$$

$$1! = 1.0! = 1.1 = 1$$

$$2! = 2.1! = 2.1 = 2$$

$$3! = 3.2! = 3.2 = 6$$

$$4! = 4.3! = 4.6 = 24$$

Recursive Algorithm for Computing $n!$

procedure *factorial*(n: nonnegative integer)
if $n = 0$ **then return** 1
else return $n \cdot factorial(n - 1)$
{output is $n!$}

Example. Using the algorithm to compute 5!

Solution.

- $5! = 5.4!$
- $4! = 4.3!$
- $3! = 3.2!$
- $2! = 2.1!$
- $1! = 1.0!$
- $0! = 1$(Basis step)
- Recursive steps

$$1! = 1$$
$$2! = 2$$
$$3! = 6$$
$$4! = 24$$
$$5! = 120$$

Recursive Algorithm for Computing a^n

procedure *power*(a: nonzero real number, n: nonnegative integer)
if $n = 0$ **then return** 1
else return $a \cdot power(a, n - 1)$
{output is a^n}

Example. Find output value if $a = 3, n = 4$

Solution.

3 Recursive Algorithms

- $3^4 = 3.3^3$
- $3^3 = 3.3^2$
- $3^2 = 3.3^1$
- $3^1 = 3.3^0$
- $3^0 = 1$ (Basis step)
- Recursive step

$$3^1 = 3$$
$$3^2 = 9$$
$$3^3 = 27$$
$$3^4 = 81$$

Recursive Algorithm for Computing gcd(a,b)

procedure *gcd*(a, b: nonnegative integers with $a < b$)
if $a = 0$ **then return** b
else return $gcd(b$ **mod** $a, a)$
{output is gcd(a, b)}

Example. Find output value if input $a = 5, b = 8$

Solution.

- $gcd(5, 8) = gcd(8 \bmod 5, 5) = gcd(3, 5)$
- $gcd(3, 5) = gcd(5 \bmod 3, 3) = gcd(2, 3)$
- $gcd(2, 3) = gcd(3 \bmod 2, 2) = gcd(1, 2)$
- $gcd(1, 2) = gcd(2 \bmod 1, 1) = gcd(0, 1)$
- return 1

Hence, $gcd(5, 8) = 1$

Recursive Modular Exponentiation

procedure *mpower*(b, n, m: integers with $b > 0$ and $m \geq 2$, $n \geq 0$)
if $n = 0$ **then**
  return 1
else if n is even **then**
  return $mpower(b, n/2, m)^2$ **mod** m
else
  return $(mpower(b, \lfloor n/2 \rfloor, m)^2 \textbf{ mod } m \cdot b \textbf{ mod } m) \textbf{ mod } m$
$\{$output is b^n **mod** $m\}$

Example. Find $2^5 \bmod 3$

Solution.

3 Recursive Algorithms

$b = 2, n = 5, m = 3$

$n = 5$ odd: $mpower(2, 5, 3) = (mpower(2, 2, 3)^2 \text{ mod } m.2 \text{ mod } m) \text{ mod } m$

$n = 2$ even: $mpower(2, 2, 3) = (mpower(2, 1, 3)^2) \text{ mod } 3$

$n = 1$ odd: $mpower(2, 1, 3) = (mpower(2, 0, 3)^2 \text{ mod } 3.2 \text{ mod } 3) \text{ mod } 3$

$mpower(2, 0, 3) = 1$ (Basis step)

Recursive steps

- $mpower(2, 1, 3) = 2$
- $mpower(2, 2, 3) = 1$
- $mpower(2, 5, 3) = 2$

Hence, $2^5 \text{ mod } 3 = 2$

Recursive Linear Search Algorithm

procedure *search*(i, j, x: integers, $1 \leq i \leq j \leq n$)
if $a_i = x$ **then**
  return i
else if $i = j$ **then**
  return 0
else
  return *search*($i + 1, j, x$)
{output is the location of x in $a_1, a_2, \ldots, a_n$ if it appears; otherwise it is 0}

Example. List all the steps used to search for 9 in the sequence $2, 3, 4, 5, 6, 8, 9, 11$

Solution.

3 Recursive Algorithms

2	3	4	5	6	8	9	11
a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8

$\rightarrow i = 1, j = 8, x = 9$

- $a_1 = 9(!) \rightarrow$ Search$(2, 8, 9)$
- $a_2 = 9(!) \rightarrow$ Search$(3, 8, 9)$
- $a_3 = 9(!) \rightarrow$ Search$(4, 8, 9)$
- $a_4 = 9(!) \rightarrow$ Search$(5, 8, 9)$
- $a_5 = 9(!) \rightarrow$ Search$(6, 8, 9)$
- $a_6 = 9(!) \rightarrow$ Search$(7, 8, 9)$
- $a_7 = 9(ok)$
- return 7

Recursive Binary Search Algorithm

procedure *binary search*$(i, j, x$: integers, $1 \le i \le j \le n)$
$m := \lfloor (i + j)/2 \rfloor$
if $x = a_m$ **then**
  return m
else if $(x < a_m$ and $i < m)$ **then**
  return *binary search*$(i, m - 1, x)$
else if $(x > a_m$ and $j > m)$ **then**
  return *binary search*$(m + 1, j, x)$
else return 0
{output is location of x in $a_1, a_2, \ldots, a_n$ if it appears; otherwise it is 0}

Example. To search for 19 in the list $1, 2, 3, 5, 6, 7, 8, 10, 12, 13, 15, 16, 18, 19, 20, 22$

Solution.

3 Recursive Algorithms

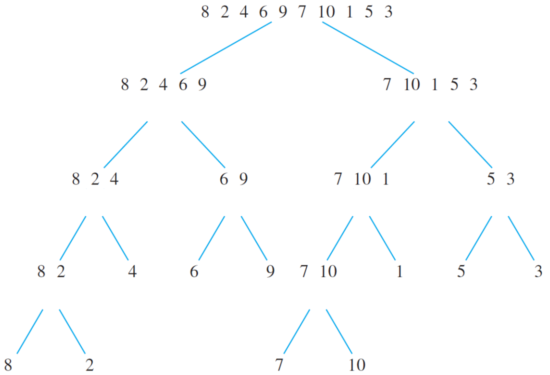
1	2	3	5	6	7	8	10	12	13	15	16	18	19	20	22
a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}	a_{11}	a_{12}	a_{13}	a_{14}	a_{15}	a_{16}

$\rightarrow i = 1, j = 16, x = 19$

- $m := \lfloor (1 + 16)/2 \rfloor = 8 : 19 > 10 \land 16 > 8 \rightarrow$ binary search$(9, 16, 19)$
- $m := \lfloor (9 + 16)/2 \rfloor = 12 : 19 > 16 \land 16 > 12 \rightarrow$ binary search$(13, 16, 19)$
- $m := \lfloor (13 + 16)/2 \rfloor = 14 : 19 = 19$
- return 14

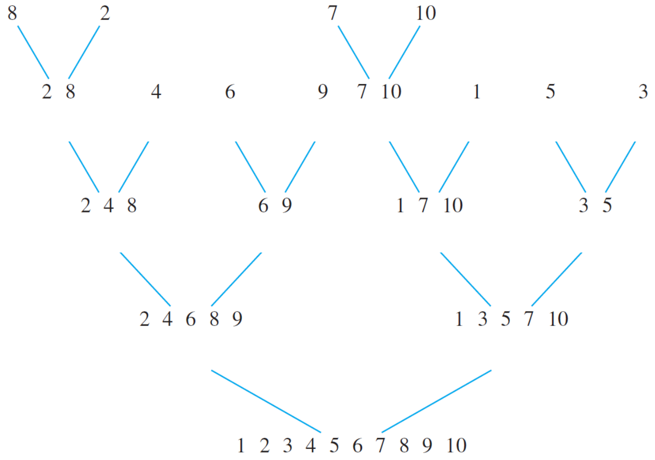
Recursive Merge Sort

Example. Use the merge sort to put the terms of the list $8, 2, 4, 6, 9, 7, 10, 1, 5, 3$ in increasing order.



Recursive Merge Sort

3 Recursive Algorithms



Recursive Merge Sort

procedure $mergesort(L = a_1, \ldots, a_n)$
if $n > 1$ **then**
    $m := \lfloor n/2 \rfloor$
    $L_1 := a_1, a_2, \ldots, a_m$
    $L_2 := a_{m+1}, a_{m+2}, \ldots, a_n$
    $L := merge(mergesort(L_1),\ mergesort(L_2))$
$\{L$ is now sorted into elements in nondecreasing order$\}$

Recursive Merge Sort two sorted lists

Example. Merging the Two Sorted Lists $2, 3, 5, 6$ and $1, 4$.

First List	*Second List*	*Merged List*	*Comparison*
2 3 5 6	1 4		$1 < 2$
2 3 5 6	4	1	$2 < 4$
3 5 6	4	1 2	$3 < 4$
5 6	4	1 2 3	$4 < 5$
5 6		1 2 3 4	
		1 2 3 4 5 6	

Recursive Merge Sort two sorted lists

procedure *merge*(L_1, L_2 : sorted lists)
$L :=$ empty list
while L_1 and L_2 are both nonempty
  remove smaller of first elements of L_1 and L_2 from its list; put it at the right end of L
  if this removal makes one list empty **then** remove all elements from the other list and
    append them to L
return L {L is the merged list with elements in increasing order}

Theorem. The number of comparisons needed to merge sort a list with n elements is $O(nlogn)$.

Table of Contents

4 Problems

1. Let P (n) be the statement that $1^2 + 2^2 + 3^2 + \ldots + n^2 = \dfrac{n(n+1)(2n+1)}{6}$ for the positive integer n.

a) What is the statement P (1)?

b) Show that P (1) is true, completing the basis step of the proof.

c) What is the inductive hypothesis?

d) What do you need to prove in the inductive step?

e) Complete the inductive step, identifying where you use the inductive hypothesis.

2. Let P (n) be the statement that $1+\frac{1}{4}+\frac{1}{9}+\ldots+\frac{1}{n^2}<2-\frac{1}{n}$, where n is an integer greater than 1.

a) What is the statement P (2)?

b) Show that P (2) is true, completing the basis step of the proof.

c) What is the inductive hypothesis?

d) What do you need to prove in the inductive step?

e) Complete the inductive step.

3. Prove the statement "6 divides n^3 - n for all integers $n \geq 0$".

4. Prove that $3^n < n!$ if n is an integer greater than 6.

5. Prove that $2^n > n^2$ if n is an integer greater than 4.

6. Prove that for every positive integer n, $1+\frac{1}{\sqrt{2}}+\frac{1}{\sqrt{3}}+\ldots+\frac{1}{\sqrt{n}}>2\sqrt{n+1}-2$.

7. Prove that $\ln n < \sum_{i=1}^{n} \frac{1}{i}$ whenever n is a positive integer.

Strong Induction

4 Problems

1. Let P (n) be the statement that a postage of n cents can be formed using just 3-cent stamps and 5-cent stamps. The parts of this exercise outline a strong induction proof that P(n) is true for $n \geq 8$.

a) Show that the statements P (8), P (9), and P (10) are true, completing the basis step of the proof.

b) What is the inductive hypothesis of the proof?

c) What do you need to prove in the inductive step?

d) Complete the inductive step for $k \geq 10$.

Strong Induction

4 Problems

2. Let P (n) be the statement that a postage of n cents can be formed using just 4-cent stamps and 7-cent stamps. The parts of this exercise outline a strong induction proof that P(n) is true for $n \geq 18$.

a) Show statements P(18), P(19), P(20), and P(21) are true, completing the basis step of the proof.

b) What is the inductive hypothesis of the proof?

c) What do you need to prove in the inductive step?

d) Complete the inductive step for $k \geq 21$.

1. Find f(1), f(2), f(3), and f(4) if f(n) is defined recursively by $f(0) = 1$ and for $n = 0, 1, \ldots$

a) $f(n + 1) = f(n) + 2$ b) $f(n + 1) = 3f(n)$

c) $f(n + 1) = 2f(n)$ d) $f(n + 1) = f(n)^2 + f(n) + 1$

2. Find f(2), f(3), f(4), and f(5) if f is defined recursively by $f(0) = -1$, $f(1) = 2$, and for $n = 1, 2, \ldots$

a) $f(n + 1) = f(n) + 3f(n - 1)$ b) $f(n + 1) = f(n)^2 f(n - 1)$

c) $f(n + 1) = 3f(n)^2 - 4f(n - 1)^2$ d) $f(n + 1) = f(n - 1)/f(n)$

3. Find $f(2)$, $f(3)$, $f(4)$, and $f(5)$ if f is defined recursively by $f(0) = f(1) = 1$ and for $n = 1, 2, \ldots$

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

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

c) $f(n+1) = f(n)2 + f(n-1)^3$

d) $f(n+1) = f(n)/f(n-1)$

4. Determine whether each of these proposed definitions is a valid recursive definition of a function f from the set of nonnegative integers to the set of integers. If f is well defined, find a formula for f(n) when n is a nonnegative integer and prove that your formula is valid.

a) $f(0) = 0$, $f(n) = 2f(n-2)$ for $n \geq 1$

b) $f(0) = 1$, $f(n) = f(n-1) - 1$ for $n \geq 1$

c) $f(0) = 2$, $f(1) = 3$, $f(n) = f(n-1) - 1$ for $n \geq 2$

5. Give a recursive definition of the sequence {an}, n = 1, 2, 3, . . . if

a) $a_n = 6n$ b) $a_n = 2n + 1$ c) $a_n = 10n$ d) $a_n = 5$

e) $a_n = 4n - 2$ f) $a_n = 1 + (-1)^n$ g) $a_n = n(n + 1)$ h) $a_n = n^2$

6. Let F be the function such that F(n) is the sum of the first n positive integers. Give a recursive definition of F(n).

7. Give a **recursive definition** of each of these sets.

a) A = {2, 5, 8, 11, 14, …} b) B = {…, -5, -1, 3, 7, 10, …}

c) C = {3, 12, 48, 192, 768, …} d) D = {1, 2, 4, 7, 11, 16, …}

8. The reversal of a string is the string consisting of the symbols of the string in reverse order. The reversal of the string w is denoted by w^R. Find the reversal of the following bit strings.

a) 0101 b) 1 1011 c) 1000 1001 0111

9. When does a string belong to the set A of bit strings defined recursively by

$\lambda \in A$, $0x1 \in A$ if $x \in A$, where λ is the empty string?

Recursive Algorithms

4 Problems

1. Give a recursive algorithm for computing nx whenever n is a positive integer and x is an integer, using just addition.

2. Consider an **recursive algorithm** to compute the n^{th} Fibonacci number:

```
procedure Fibo(n : positive integer)
if n = 1 return 1
else    if n = 2 return 1
        else return Fibo(n – 1) + Fibo(n – 2)
```

How many additions (+) are used to find Fibo(6) by the algorithm above?

3. Give a recursive algorithm for finding the sum of the first n odd positive integers.

4. Consider the following algorithm:

    procedure tinh(a: real number; n: positive integer)

    if n = 1 return a

    else return a·tinh(a, n-1).

a) What is the output if inputs are: n = 4, a = 2.5? Explain your answer.

b) Show that the algorithm computes n·a using Mathematical Induction.

5. Consider the following algorithm:

procedure $F(a_1, a_2, a_3, \ldots, a_n$: integers)

if $n = 0$ return 0

else return $a_n + F(a_1, a_2, a_3, \ldots, a_{n-1})$

Find

a) $F(1,3,5)$ b) $F(1,3,5)$ c) $F(1,2,3,5,9)$

Q&A

Thank you for listening!