Induction and Recursion

Chapter 5

With Question/Answer Animations

Because learning changes everything.™

1

Chapter Summary

- Mathematical Induction
- Strong Induction
- Well-Ordering
- Recursive Definitions
- Structural Induction
- Recursive Algorithms
- Program Correctness (*not yet included in overheads*)

2

Mathematical Induction

- Section 5.1

3

Section Summary [1]

- Mathematical Induction
- Examples of Proof by Mathematical Induction
- Mistaken Proofs by Mathematical Induction
- Guidelines for Proofs by Mathematical Induction

4

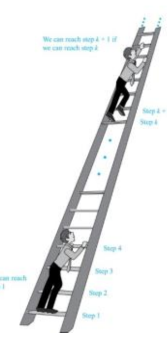
Climbing an Infinite Ladder

Suppose we have an infinite ladder:

1. We can reach the first rung of the ladder.
2. If we can reach a particular rung of the ladder, then we can reach the next rung.

From (1), we can reach the first rung. Then by applying (2), we can reach the second rung. Applying (2) again, the third rung. And so on. We can apply (2) any number of times to reach any particular rung, no matter how high up.

This example motivates proof by mathematical induction.



Jump to long description

5

Principle of Mathematical Induction

- *Principle of Mathematical Induction*: To prove that $P(n)$ is true for all positive integers n, we complete these steps:
  - *Basis Step*: Show that $P(1)$ is true.
  - *Inductive Hypothesis*: Assume that $P(k)$ is true.
  - *Inductive Step*: Show that $P(k+1)$ is true for all positive integers k.
- To complete the inductive step, assuming the *inductive hypothesis* that $P(k)$ holds for an arbitrary integer k, show that $P(k+1)$ be true.
- **Climbing an Infinite Ladder Example**:
  - BASIS STEP: We can reach rung 1
  - INDUCTIVE HYPOTHESIS: Assume that we can reach rung k
  - INDUCTIVE STEP: If we can reach rung k, then we can reach rung $k+1$.

Hence, we can reach every rung on the ladder.

6

Important Points About Using Mathematical Induction

Mathematical induction can be expressed as the rule of inference

$$\big(P(1) \wedge \forall k\,(P(k) \rightarrow P(k+1))\big) \rightarrow \forall n\, P(n),$$

where the domain is the set of positive integers.

In a proof by mathematical induction, we don't assume that $P(k)$ is true for all positive integers! We show that if we assume that $P(k)$ is true for some k, then $P(k + 1)$ must also be true.

Proofs by mathematical induction do not always start at the integer 1. In such a case, the basis step begins at a starting point b where b is an integer. We will see examples of this soon.

7

Validity of Mathematical Induction

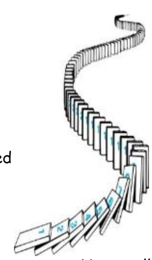
- Mathematical induction is valid because of the well ordering property, which states that every nonempty subset of the set of positive integers has a least element (*see Section* 5.2 *and Appendix* 1). Here is the proof:
  - Suppose that $P(1)$ holds and $P(k) \rightarrow P(k + 1)$ is true for all positive integers k.
  - Assume there is at least one positive integer n for which $P(n)$ is false. Then the set S of positive integers for which $P(n)$ is false is nonempty.
  - By the well-ordering property, S has a least element, say m.
  - We know that m can not be 1 since $P(1)$ holds.
  - Since m is positive and greater than 1, $m - 1$ must be a positive integer. Since $m - 1 < m$, it is not in S, so $P(m - 1)$ must be true.
  - But then, since the conditional $P(k) \rightarrow P(k + 1)$ for every positive integer k holds, $P(m)$ must also be true. This contradicts $P(m)$ being false.
  - Hence, $P(n)$ must be true for every positive integer n.

8

Remembering How Mathematical Induction Works

Consider an infinite sequence of dominoes, labeled 1,2,3, …, where each domino is standing.

Let $P(n)$ be the proposition that the nth domino is knocked over.



We know that the first domino is knocked down, i.e., $P(1)$ is true .

We also know that if whenever the kth domino is knocked over, it knocks over the $(k + 1)$st domino, i.e, $P(k) \rightarrow P(k + 1)$ is true for all positive integers k.

Hence, all dominos are knocked over.

$P(n)$ is true for all positive integers n.

Jump to long description

9

Proving a Summation Formula by Mathematical Induction

Example: Show that: $\displaystyle\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$

Solution:

Note: Once we have this conjecture, mathematical induction can be used to prove it correct.

- BASIS: $n = 1$

  lhs: $\displaystyle\sum_{i=1}^{1} i = 1$ rhs: $\displaystyle\frac{1(1+1)}{2} = 1$ ✓

- INDUCTIVE HYPOTHESIS: Assume $\displaystyle\sum_{i=1}^{k} i = \frac{k(k+1)}{2}$ for some k

- INDUCTIVE STEP: Show: $\displaystyle\sum_{i=1}^{k+1} i = \frac{(k+1)(k+2)}{2}$

$$\sum_{i=1}^{k+1} i = \sum_{i=1}^{k} i + (k+1)$$

$$= \frac{k(k+1)}{2} + (k+1) \quad \text{by I.H.}$$

$$= \frac{k(k+1) + 2(k+1)}{2}$$

$$= \frac{(k+1)(k+2)}{2}$$

10

Conjecturing and Proving Correct a Summation Formula

Example: Conjecture and prove correct a formula for the sum of the first n positive odd integers. Then prove your conjecture.

Solution: We have: $1 = 1,\ 1 + 3 = 4,\ 1 + 3 + 5 = 9,\ 1 + 3 + 5 + 7 = 16,\ 1 + 3 + 5 + 7 + 9 = 25.$

We can conjecture that the sum of the first n positive odd integers is n^2,

$$1 + 3 + 5 + \cdots + (2n - 1) = n^2.$$

Inductive Proof:

BASIS: $n = 1$

lhs: 1 rhs: $1^2 = 1$ ✓

INDUCTIVE HYPOTHESIS: Assume $1 + 3 + 5 + \cdots + (2k - 1) = k^2$ for some k

INDUCTIVE STEP: Show: $1 + 3 + 5 + \cdots + (2k - 1) + (2k + 1) = (k + 1)^2$

$$1 + 3 + 5 + \cdots + (2k - 1) + (2k + 1) = k^2 + 2k + 1 \quad \text{by I.H.}$$

$$= (k + 1)^2$$

By induction, we have shown the original statement to be true.

11

Classroom Exercise

Prove: $1^2 + 3^2 + 5^2 + \cdots + (2n + 1)^2 = \frac{(n+1)(2n+1)(2n+3)}{3}$ whenever n is a nonnegative integer.

12

Proving Inequalities[1]

Example: Use mathematical induction to prove that $n < 2^n$ for all positive integers *n*.

Solution: Let $P(n)$ be the proposition that $n < 2^n$.

- BASIS: $P(1)$ is true since lhs: 1, rhs: $2^1 = 2$, and $1 < 2$ ✓
- INDUCTIVE HYPOTHESIS: Assume $k < 2^k$, for an arbitrary positive integer k
- INDUCTIVE STEP: Show: $k + 1 < 2^{k+1}$

$$\begin{aligned} k + 1 &< 2^k + 1 \quad \text{by I.H.} \\ &\le 2^k + 2^k \\ &= 2^{k+1} \end{aligned}$$

Therefore $n < 2^n$ holds for all positive integers *n*.

13

Proving Inequalities[2]

Example: Use induction to prove that $2^n < n!$, for $n \ge 4$.

Solution: Let $P(n)$ be the proposition that $2^n < n!$

- BASIS: $n = 4$ lhs: $2^4 = 16$ and rhs: $4! = 24$, and $16 < 24$ ✓
- INDUCTIVE HYPOTHESIS: Assume $2^k < k!$ for an arbitrary integer $k \ge 4$
- INDUCTIVE STEP: Show $(k + 1)$: $2^{k+1} < (k + 1)!$

$$\begin{aligned} 2^{k+1} &= 2 \cdot 2^k \\ &< 2 \cdot k! \qquad \text{(by the inductive hypothesis)} \\ &< (k+1)k! \\ &= (k+1)! \end{aligned}$$

Therefore, $2^n < n!$ holds, for every integer $n \ge 4$.

Note that here the basis step is $P(4)$, since $P(0)$, $P(1)$, $P(2)$, and $P(3)$ are all false.

14

Classroom Exercise

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

15

Proving Divisibility Results

Example: Use mathematical induction to prove that $n^3 - n$ is divisible by 3, for every positive integer *n*.

Solution: Let $P(n)$ be the proposition that $n^3 - n$ is divisible by 3.

- BASIS: $P(1)$ is true since $1^3 - 1 = 0$, which is divisible by 3.
- INDUCTIVE HYPOTHESIS: Assume $P(k)$: $k^3 - k$ is divisible by 3, for an arbitrary positive integer k.
- INDUCTIVE STEP: Show: $P(k + 1)$: $(k + 1)^3 - (k + 1)$ is divisible by 3

$$\begin{aligned} (k+1)^3 - (k+1) &= (k^3 + 3k^2 + 3k + 1) - (k+1) \\ &= (k^3 - k) + 3(k^2 + k) \end{aligned}$$

By the inductive hypothesis, the first term $(k^3 - k)$ is divisible by 3 and the second term is divisible by 3 since it is an integer multiplied by 3. So by part (i) of Theorem 1 in Section 4.1 , $(k + 1)^3 - (k + 1)$ is divisible by 3.

Therefore, $n^3 - n$ is divisible by 3, for every integer positive integer *n*.

16

Number of Subsets of a Finite Set

Example: Use mathematical induction to show that if *S* is a finite set with n elements, where *n* is a nonnegative integer, then *S* has 2^n subsets.

(*Chapter 6 uses combinatorial methods to prove this result.*)

Solution: Let $P(n)$ be the proposition that a set with *n* elements has 2^n subsets.

- BASIS: $P(0)$ is true, because the empty set has only itself as a subset and $2^0 = 1$.
- INDUCTIVE HYPTHESIS: Assume $P(k)$: every set with k elements has 2^k subsets for an arbitrary nonnegative integer k.

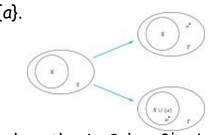
17

Number of Subsets of a Finite Set

INDUCTIVE STEP: Show that a set with $k + 1$ elements has 2^{k+1} subsets.

Let T be a set with $k + 1$ elements. Then $T = S \cup \{a\}$, where $a \in T$ and $S = T - \{a\}$. Hence $|S| = k$.

For each subset X of S, there are exactly two subsets of T, i.e., X and $X \cup \{a\}$.



By the inductive hypothesis S has 2^k subsets. Since there are two subsets of T for each subset of S, the number of subsets of T is $2 \cdot 2^k = 2^{k+1}$.

Jump to long description

18

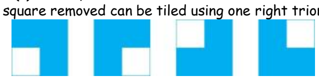
Tiling Checkerboards

Example: Show that every $2^n \times 2^n$ checkerboard with one square removed can be tiled using right triominoes.

A right triomino is an L-shaped tile which covers three squares at a time.

Solution: Let $P(n)$ be the proposition that every $2^n \times 2^n$ checkerboard with one square removed can be tiled using right triominoes. Use mathematical induction to prove that $P(n)$ is true for all positive integers n.

- BASIS : P(1) is true, because each of the four 2×2 checkerboards with one square removed can be tiled using one right triomino.



- INDUCTIVE HYPOTHESIS: Assume that $P(k)$ is true: every $2^k \times 2^k$ checkerboard with one square removed can be tiled using right triominoes for some positive integer k.

Jump to long description

19

Tiling Checkerboards

INDUCTIVE STEP: Show: Consider a $2^{k+1} \times 2^{k+1}$ checkerboard with one square missing can be tiled with right triominoes.

Consider a $2^{k+1} \times 2^{k+1}$ checkerboard with one square removed. Split this checkerboard into four checkerboards of size $2^k \times 2^k$,by dividing it in half in both directions.

Remove a square from one of the four $2^k \times 2^k$ checkerboards. By the inductive hypothesis, this board can be tiled. Also by the inductive hypothesis, the other three boards can be tiled with the square from the corner of the center of the original board removed. We can then cover the three adjacent squares with a triomino.

Hence, the entire $2^{k+1} \times 2^{k+1}$ checkerboard with one square removed can be tiled using right triominoes.

Jump to long description

20

Guidelines: Mathematical Induction Proofs

Template for Proofs by Mathematical Induction

1. Express the statement that is to be proved in the form "for all $n \geq b$, $P(n)$" for a fixed integer b.
2. Write out "BASIS STEP." Then show that $P(b)$ is true, taking care that the correct value of b is used, with the left- and right-hand sides computed independently.
3. Write out "INDUCTIVE HYPOTHESIS". State, and clearly identify, the inductive hypothesis, in the form "assume that $P(k)$ is true for an arbitrary fixed integer $k \geq b$."
4. Write out "INDUCTIVE STEP." State what needs to be shown under the assumption that the inductive hypothesis is true, i.e., write out $P(k + 1)$.
5. Prove the statement $P(k + 1)$ making use of the assumption $P(k)$. Be sure that your proof is valid for all integers k with $k \geq b$, taking care that the proof works for small values of k, including $k = b$.
6. Clearly identify the conclusion of the inductive step where you reach the conclusion in the "Show" statement.
7. After completing the basis, inductive hypothesis, and inductive step, state the conclusion, namely, by mathematical induction, $P(n)$ is true for all integers n with $n \geq b$.

21

Strong Induction and Well-Ordering

- Section 5.2

22

Section Summary 2

- Strong Induction
- Example Proofs using Strong Induction
- Well-Ordering Property

23

Strong Induction

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

- *Basis Step*: Verify that the proposition $P(1)$ is true.
- *Inductive Hypothesis*: Assume $P(j)$ holds for all $j \leq k$
- *Inductive Step*: Show the conditional statement holds for all positive integers k.

$$[P(1) \wedge P(2) \wedge \cdots \wedge P(k)] \rightarrow P(k+1)$$

Strong Induction is sometimes called the *second principle of mathematical induction* or *complete induction.*

24

Strong Induction and the Infinite Ladder

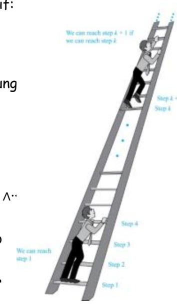
Strong induction tells us that we can reach all rungs if:

1. We can reach the first rung of the ladder.
2. Assume we can reach the first k rungs.
3. For every integer k, if we can reach the first k rung then we can reach the $(k + 1)$st rung.

To conclude that we can reach every rung by strong induction:

- BASIS STEP: $P(1)$ holds
- INDUCTIVE HYPOTHESIS: Assume $P(1) \wedge P(2) \wedge \cdots P(k)$ holds for an arbitrary integer k
- INDUCTIVE STEP: Show that $P(k + 1)$ must also hold.

We will have then shown by strong induction that for every positive integer n, $P(n)$ holds, i.e., we can reach the nth rung of the ladder.



25

Proof Using Strong Induction

- **Example**: Suppose we can reach the first and second rungs of an infinite ladder, and we know that if we can reach a rung, then we can reach two rungs higher. Prove that we can reach every rung. (Try this with mathematical induction.)
- **Solution**: Prove the result using strong induction.
- BASIS STEP: We can reach the first step.
- INDUCTIVE HYPOTHESIS: Assume that we can reach the first j rungs for any $2 \leq j \leq k$.
- INDUCTIVE STEP: Show that we can reach the $(k + 1)$st rung.
- Hence, we can reach all rungs of the ladder.

26

Which Form of Induction Should Be Used?

- We can always use strong induction instead of mathematical induction. But there is no reason to use it if it is simpler to use mathematical induction.
- Regular induction is just a special case of strong induction.
- In fact, the principles of mathematical induction, strong induction, and the well-ordering property are all equivalent.
- Sometimes it is clear how to proceed using one of the three methods, but not the other two.

27

Proof of the Fundamental Theorem of Arithmetic

- **Example**: Show that if n is an integer greater than 1, then n can be written as the product of primes.
- **Solution**: Let $P(n)$ be the proposition that n can be written as a product of primes.
- BASIS STEP: $P(2)$ is true since 2 itself is prime.
- INDUCTIVE HYPOTHESIS: Assume $P(j)$ is true for $2 \leq j \leq k$.
- INDUCTIVE STEP: Show that $P(k + 1)$ must be true under this assumption; two cases need to be considered:
  - if $k + 1$ is prime, then $P(k + 1)$ is true
  - otherwise, $k + 1$ is composite and can be written as the product of two positive integers a and b with $2 \leq a \leq b < k + 1$
    - by the inductive hypothesis a and b can be written as the product of primes and therefore $k + 1$ can also be written as the product of those primes
- Hence, every integer > 1 can be written as the product of primes.

28

Proof Using Strong Induction

- **Example**: Prove that every amount of postage of 12 cents or more can be formed using just 4-cent and 5-cent stamps.
- **Solution**: Let $P(n)$ be the proposition that postage of n cents can be formed using 4-cent and 5-cent stamps.
- BASIS STEP: $P(12)$, $P(13)$, $P(14)$, and $P(15)$ hold.
  - $P(12)$ uses three 4-cent stamps.
  - $P(13)$ uses two 4-cent stamps and one 5-cent stamp.
  - $P(14)$ uses one 4-cent stamp and two 5-cent stamps.
  - $P(15)$ uses three 5-cent stamps.
- INDUCTIVE HYPOTHESIS: Assume that $P(j)$ holds for $12 \leq j \leq k$, where $k \geq 15$.
- INDUCTIVE STEP: Show that $P(k + 1)$ holds.
  - Using the inductive hypothesis, $P(k - 3)$ holds since $k - 3 \geq 12$. To form postage of $k + 1$ cents, add a 4-cent stamp to the postage for $k - 3$ cents. Hence, $P(n)$ holds for all $n \geq 12$.

29

Proof of Same Example using Mathematical Induction

- **Example**: Prove that every amount of postage of 12 cents or more can be formed using just 4-cent and 5-cent stamps.
- **Solution**: Let $P(n)$ be the proposition that postage of n cents can be formed using 4-cent and 5-cent stamps.
- BASIS STEP: Postage of 12 cents can be formed using three 4-cent stamps.
- INDUCTIVE HYPOTHESIS: Assume that postage of k cents can be formed using 4-cent and 5-cent stamps for some k
- INDUCTIVE STEP: Show: $P(k + 1)$ where $k \geq 12$
  - We consider two cases:
    - If at least one 4-cent stamp has been used, then a 4-cent stamp can be replaced with a 5-cent stamp to yield a total of $k + 1$ cents.
    - Otherwise, no 4-cent stamp have been used and at least three 5-cent stamps were used. Three 5-cent stamps can be replaced by four 4-cent stamps to yield a total of $k + 1$ cents.
- Hence, $P(n)$ holds for all $n \geq 12$.

30

Well-Ordering Property

- *Well-ordering property*: Every nonempty set of nonnegative integers has a least element.
- The well-ordering property is one of the axioms of the positive integers listed in Appendix 1.
- The well-ordering property can be used directly in proofs
- The well-ordering property can be generalized.
- **Definition:** A set is *well ordered* if every subset has a least element.
  - **N** is well ordered under ≤.
  - The set of finite strings over an alphabet using lexicographic ordering is well ordered.
- We will see a generalization of induction to sets other than the integers in the next section.

31

Recursive Definitions and Structural Induction

- Section 5.3

32

Section Summary

- Recursively Defined Functions
- Recursively Defined Sets and Structures
- Structural Induction
- Generalized Induction

33

Recursively Defined Functions

- **Definition:** A *recursive* or *inductive definition* of a function consists of two steps.
  - BASIS STEP: Specify the value of the function at zero.
  - RECURSIVE STEP: Give a rule for finding its value at an integer from its values at smaller integers.
- A function $f(n)$ is the same as a sequence $a_0, a_1, \ldots$, where a_i, where $f(i) = a_i$. This was done using recurrence relations in Section 2.4.

34

Recursively Defined Functions

Example: Suppose f is defined by:

$$f(0) = 3,$$
$$f(n+1) = 2f(n) + 3$$

Find $f(1)$, $f(2)$, $f(3)$, $f(4)$

Solution:

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

Example: Give a recursive definition of the factorial function $n!$

Solution: $f(0) = 1$

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

35

Recursively Defined Functions

Example: Give a recursive definition of:

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

Solution: The first part of the definition is

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

The second part is $$\sum_{k=0}^{n+1} a_k = \left(\sum_{k=0}^{n} a_k\right) + a_{n+1}$$

36

Fibonacci Numbers


Fibonacci
(1170- 1250)

Example : The Fibonacci numbers are defined as follows:

$$f_0 = 0$$
$$f_1 = 1$$
$$f_n = f_{n-1} + f_{n-2}$$

Find f_2, f_3, f_4, f_5.

$$f_2 = f_1 + f_0 = 1 + 0 = 1$$
$$f_3 = f_2 + f_1 = 1 + 1 = 2$$
$$f_4 = f_3 + f_2 = 2 + 1 = 3$$
$$f_5 = f_4 + f_3 = 3 + 2 = 5$$

In Chapter 8, we will use the Fibonacci numbers to model population growth of rabbits. This was an application described by Fibonacci himself.

Next, we use strong induction to prove a result about the Fibonacci numbers.

37

Recursively Defined Sets and Structures

- *Recursive definitions* of sets have two parts:
  - The *basis step* specifies an initial collection of elements.
  - The *recursive step* gives the rules for forming new elements in the set from those already known to be in the set.
- Sometimes the recursive definition has an *exclusion rule*, which specifies that the set contains nothing other than those elements specified in the basis step and generated by applications of the rules in the recursive step.
- We will always assume that the exclusion rule holds, even if it is not explicitly mentioned.
- We will later develop a form of induction, called *structural induction*, to prove results about recursively defined sets.

38

Recursively Defined Sets and Structures

- **Example**: integers that are positive multiples of 3

  BASIS STEP: 3 ∊ S.

  RECURSIVE STEP: If x ∊ S, then x + 3 ∊ S

  Initially 3 is in S, then 3 + 3 = 6, then 3 + 6 = 9, etc.
- **Example**: the natural numbers **N**

  BASIS STEP: 1 ∊ **N**.

  RECURSIVE STEP: If n is in **N**, then n + 1 is in **N**.

  Initially 1 is in S, then 1 + 1 = 2, then 1 + 2 = 3, etc.

39

Strings

- **Definition**: The set Σ* of *strings* over the alphabet Σ:

  BASIS STEP: λ ∊ Σ* (λ is the empty string)

  RECURSIVE STEP: If *w* is in Σ* and *x* is in Σ, then *wx* ∊ Σ*.
- **Example**: If Σ = {0,1}, the strings in in Σ* are the set of all bit strings, λ, 0, 1, 00, 01, 10, 11, etc.
- **Example**: If Σ = {*a*,*b*}, show that *aab* is in Σ*.

  since λ ∊ Σ* and *a* ∊ Σ, *a* ∊ Σ*

  since *a* ∊ Σ* and *a* ∊ Σ, *aa* ∊ Σ*

  since *aa* ∊ Σ* and *b* ∊ Σ, *aab* ∊ Σ*

40

String Concatenation

- **Definition**: Two strings can be combined via the operation of *concatenation*. Let Σ be a set of symbols and Σ* be the set of strings formed from the symbols in Σ. We can define the concatenation of two strings, denoted by ∙, recursively as follows.

  BASIS STEP: If *w* ∊ Σ*, then *w* ∙ λ= *w*.

  RECURSIVE STEP: If w_1 ∊ Σ* and w_2 ∊ Σ* and *x* ∊ Σ, then $w_1 \cdot (w_2\, x) = (w_1 \cdot w_2)x$.

  often $w_1 \cdot w_2$ is written as $w_1\, w_2$

  if w_1 = *abra* and w_2 = *cadabra*, the concatenation $w_1\, w_2$ = *abracadabra*

41

Length of a String

- **Example**: Give a recursive definition of $l(w)$, the length of the string *w*.
- **Solution**: The length of a string can be recursively defined by:

$$l(\lambda) = 0;$$
$$l(wx) = l(w) + 1 \text{ if } w \in \Sigma^* \text{ and } x \in \Sigma.$$

42

Balanced Parentheses

- **Example**: Give a recursive definition of the set of balanced parentheses P.
- **Solution**:

  BASIS STEP: () ∈ P

  RECURSIVE STEP: If $w \in P$, then () $w \in P$, (w) ∈ P and w () ∈ P
- Show that (() ()) is in P.
- Why is))(() not in P?

43

Well-Formed Formulae in Propositional Logic

- **Definition**: The set of *well-formed formulae* in propositional logic involving **T**, **F**, propositional variables, and operators from the set {¬,∧,∨,→,↔}.

BASIS STEP: **T**, **F**, and s, where s is a propositional variable, are well-formed formulae.

RECURSIVE STEP: If E and F are well formed formulae, then $(\neg E)$, $(E \wedge F)$, $(E \vee F)$, $(E \rightarrow F)$, $(E \leftrightarrow F)$, are well-formed formulae.

- **Examples**: $((p \vee q) \rightarrow (q \wedge \mathbf{F}))$ is a well-formed formula.
  - $pq \wedge$ is not a well formed formula

44

Rooted Trees

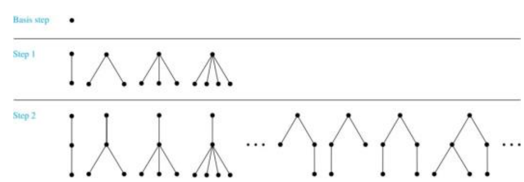
- **Definition**: The set of *rooted trees*, where a rooted tree consists of a set of vertices containing a distinguished vertex called the *root*, and edges connecting these vertices, can be defined recursively by these steps:

BASIS STEP: A single vertex r is a rooted tree.

RECURSIVE STEP: Suppose that $T_1, T_2, \ldots, T_n$ are disjoint rooted trees with roots $r_1, r_2, \ldots, r_n$, respectively. Then the graph formed by starting with a root r, which is not in any of the rooted trees $T_1, T_2, \ldots, T_n$, and adding an edge from r to each of the vertices $r_1, r_2, \ldots, r_n$, is also a rooted tree.

45

Building Up Rooted Trees



Trees are studied extensively in Chapter 11.

Next we look at a special type of tree, the full binary tree.

Jump to long description

46

Full Binary Trees

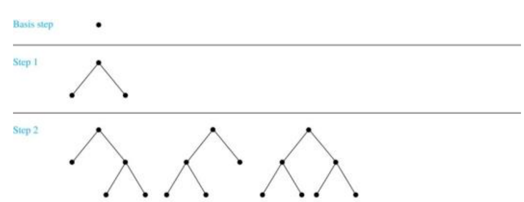
- **Definition:** The set of *full binary trees* can be defined recursively by these steps.

BASIS STEP: There is a full binary tree consisting of only a single vertex r.

RECURSIVE STEP: If T_1 and T_2 are disjoint full binary trees, there is a full binary tree, denoted by $T_1 \cdot T_2$, consisting of a root r together with edges connecting the root to each of the roots of the left subtree T_1 and the right subtree T_2.

47

Building Up Full Binary Trees



Jump to long description

48

Structural Induction

- **Definition**: To prove a property of the elements of a recursively defined set, we use *structural induction*.

  BASIS STEP: Show that the result holds for all elements specified in the basis step of the recursive definition.

  RECURSIVE STEP: Show that if the statement is true for each of the elements used to construct new elements in the recursive step of the definition, the result holds for these new elements.
- The validity of structural induction can be shown to follow from the principle of mathematical induction.

49

Full Binary Trees

Definition: The *height* $h(T)$ of a full binary tree T is defined recursively as follows:

- **BASIS STEP**: The height of a full binary tree T consisting of only a root r is $h(T) = 0$.
- **RECURSIVE STEP**: If T_1 and T_2 are full binary trees, then the full binary tree $T = T_1 \cdot T_2$ has height

$$h(T) = 1 + \max(h(T_1), h(T_2)).$$

The number of vertices $n(T)$ of a full binary tree T satisfies the following recursive formula:

- **BASIS STEP**: The number of vertices of a full binary tree T consisting of only a root r is $n(T) = 1$.
- **RECURSIVE STEP**: If T_1 and T_2 are full binary trees, then the full binary tree $T = T_1 \cdot T_2$ has the number of vertices

$$n(T) = 1 + n(T_1) + n(T_2).$$

50

Structural Induction and Binary Trees

Theorem: If T is a full binary tree, then $n(T) \leq 2^{h(T)+1} - 1$

Proof: Use structural induction.

BASIS STEP: Consider a full binary tree consisting only of a root, where $n(T) = 1$ and $h(T) = 0$ lhs: 1 and rhs: $2^{0+1} - 1$ $1 \leq 1$ ✓

INDUCTIVE HYPOTHESIS: Assume $n(T) \leq 2^{h+1} - 1$ for any tree T of height h

INDUCTIVE STEP: Show: $n(T) \leq 2^{h+2} - 1$ for tree of height h+1
we can create a new tree T' of height h+1 by adding a new root with two children of height at most h:

$$\begin{aligned} n(T') &\leq 2^{h+1} - 1 + 2^{h+1} - 1 + 1 \quad \text{by I.H.} \\ &= 2 \cdot 2^{h+1} - 1 \\ &= 2^{h+2} - 1 \ ✓ \end{aligned}$$

51

Recursive Algorithms

- Section 5.4

52

Section Summary[4]

- Recursive Algorithms
- Proving Recursive Algorithms Correct
- Merge Sort

53

Recursive Algorithms

- **Definition**: An algorithm is called *recursive* if it solves a problem by reducing it to an instance of the same problem with smaller input.
- For the algorithm to terminate, the instance of the problem must eventually be reduced to some initial case for which the solution is known.

54

Recursive Factorial Algorithm

Example: Give a recursive algorithm for computing $n!$, where n is a nonnegative integer.

Solution: Use the recursive definition of the factorial function.

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

55

Recursive Exponentiation Algorithm

Example: Give a recursive algorithm for computing a^n, where a is a nonzero real number and n is a nonnegative integer.

Solution: Use the recursive definition of 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}

56

Recursive GCD Algorithm

Example: Give a recursive algorithm for computing the greatest common divisor of two nonnegative integers a and b with $a < b$.

Solution: Use the reduction

$\gcd(a,b) = \gcd(b \textbf{ mod } a, a)$

and the condition $\gcd(0,b) = b$ when $b > 0$.

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

57

Recursive Binary Search Algorithm

Example: Construct a recursive version of a binary search algorithm.

Solution: Assume we have $a_1, a_2, \ldots, a_n$, an increasing sequence of integers. Initially i is 1 and j is n. We are searching for x.

> **procedure** *binary search*(i, j, x : integers, $1 \leq i \leq j \leq n$)
> $m := \lfloor (i + j)/2 \rfloor$
> **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 0}

58

Merge Sort

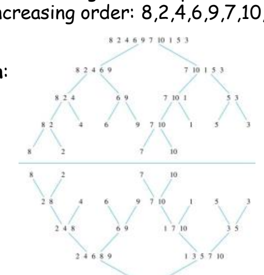
- *Merge Sort* works by iteratively splitting a list (with an even number of elements) into two sublists of equal length until each sublist has one element.
- Each sublist is represented by a balanced binary tree.
- At each step a pair of sublists is successively merged into a list with the elements in increasing order. The process ends when all the sublists have been merged.
- The succession of merged lists is represented by a binary tree.

59

Merge Sort

Example: Use merge sort to put the following list in increasing order: 8,2,4,6,9,7,10, 1, 5, 3

Solution:



Jump to long description

60

Recursive Merge Sort

Example: Construct a recursive merge sort algorithm.

Solution: Begin with the list of n elements L.

procedure *mergesort*($L = a_1, a_2,\ldots,a_n$)
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 increasing order}

61

Recursive Merge Sort

Subroutine *merge*, which merges 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 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 the elements in increasing order}

Complexity of Merge: Two sorted lists with m elements and n elements can be merged into a sorted list using no more than $m + n - 1$ comparisons.

62

Merging Two Lists

Example: Merge the two lists 2,3,5,6 and 1,4.

Solution:

TABLE 1 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	

63

Appendix of Image Long Descriptions

64

Climbing an Infinite Ladder- Appendix

- There is a man climbing an infinite ladder, the steps of which are numbered with natural numbers from the bottom. The man can reach step k plus one if he can reach step k.

Jump to the image

65

Remembering How Mathematical Induction Works- Appendix

- There are dominoes numbered with natural numbers. The domino with number one falls on the domino with number two, and the domino with number two falls on the domino with number three, etc.

Jump to the image

66

Number of Subsets of a Finite Set₂- Appendix

- There is field S with circle X inside. S has two arrows. The first one is from S to field T that has circle X and element A inside, the second arrow is from S to field T that has a circle named X union left brace A right brace inside, which has element A inside.

Jump to the image

67

Tiling Checkerboards₁- Appendix

- There are four checkerboards of the size 2 times 2 with one square removed each. The first checkerboard does not have the left bottom square. The second checkerboard does not have the right bottom square. The third checkerboard does not have the left top square, and the fourth one does not have the right top square. In each case, the remaining squares form right triomino.

Jump to the image

68

Tiling Checkerboards₂- Appendix

- There are four squares forming together a large square. The right bottom square has a small shaded square inside.

- There are four squares forming together a large square, each one has small shaded square inside. In the left top square, the small square is in the bottom right corner. In the right top square, the small square is in the bottom left corner. In the left bottom square, the small square is in the top right corner. Thus, these three small squares form a right triomino.

Jump to the image

69

Building Up Rooted Trees - Appendix

- There are 3 steps of building up rooted trees shown. Basic step contains one vertex which is a root. The first step contains several vertices that are added to the next level. All these vertices are connected to the root. At the second step, the vertices of the previous level are the roots for the added vertices of the next level.

Jump to the image

70

Building Up Full Binary Trees - Appendix

- There are three steps of building up full binary trees shown. The basic step contains one vertex on the first level which is the root. Each next level is located below the previous one. At the first step, two vertices are added to the next level. They are connected to the root forming the right and left branches. At the second step, the vertices of the second level are the roots for the added vertices of the third level. The right and the left branches can be formed in several ways: only from the left vertex, only from the right, or from both vertices.

Jump to the image

71

Merge Sort₂ - Appendix

- There is a binary balanced tree at the top. Its root consists of numbers 8, 2, 4, 6, 9, 7, 10, 1, 5, and 3. At the first step, there are two branches, the left leads to the vertex containing elements 8, 2, 4, 6, and 9. The right one leads to the vertex containing elements 7, 10, 1, 5, and 3. Two branches lead from the vertex 8, 2, 4, 6, 9 . The left one leads to the vertex with the elements 8, 2, 4. The right one leads to the vertex with the elements 6 and 9. Two branches also lead from the vertex 7, 10, 1, 5, and 3. The left one leads to the vertex with the elements 7, 10, 1. The right one leads to the vertex with the elements 5 and 3. Each of the four vertices of the previous level has two branches leading to the vertices of the next level: from 8, 2, 4 to 8, 2 and 4, from 6, 9 to 6 and 9, from 7, 10, 1 to 7, 10 and 1, from 5, 3 to 5 and 3. At the next level there are branches from 8, 2 to 8 and 2, and from 7, 10 to 7 and 10. At the bottom of the picture there is a similar tree, but it is turned upside down. In such vertices where there is more than one element, elements are written in the increasing order from left to right.

Jump to the image

72