



Discrete Structures

(CKC111)

Week 11 & Week 12



Induction and Recursion

Section Topics

Induction and Recursion

- 1 Mathematical & Strong Induction
- 2 Recursive Definitions and Structural Induction
- 3 Recursive Algorithms

Induction and Recursion



Mathematical & Strong Induction

Section Summary

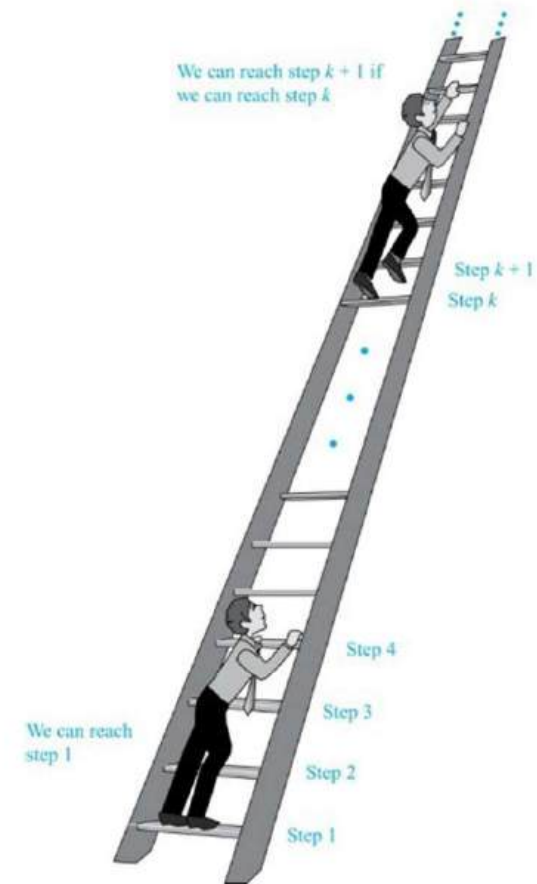


- ✓ Mathematical Induction
- ✓ Examples of Proof by Mathematical Induction
- ✓ Guidelines for Proofs by Mathematical Induction
- ✓ Strong Induction
- ✓ Example Proofs using Strong Induction

Mathematical Induction

Climbing an Infinite Ladder

- Suppose we have an infinite ladder:
- We can reach the first rung of the ladder.
- 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.



Mathematical Induction

PRINCIPLE OF MATHEMATICAL INDUCTION 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 $P(k) \rightarrow P(k + 1)$ is true for all positive integers k . $P(k)$ is called inductive hypothesis.

BASIS STEP: $P(1)$ is true.

INDUCTIVE STEP: For all positive integers k , if $P(k)$ is true, then $P(k + 1)$ is true.

$$\forall k(P(k) \rightarrow P(k + 1))$$

After completing the basis and inductive steps of a proof that $P(n)$ is true for all positive integers n . Expressed as a rule of inference, this proof technique can be stated as

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

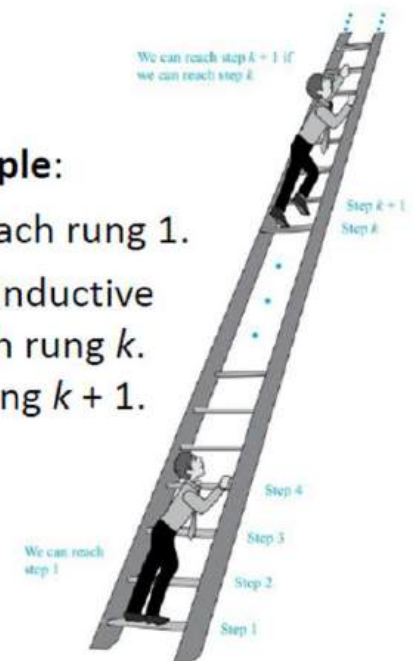
when the domain is the set of positive integers

Principle of Mathematical Induction

Example:

Climbing an Infinite Ladder Example:

- **Basis Step:** By (1), we can reach rung 1.
- **Inductive Step:** Assume the inductive hypothesis that we can reach rung k . Then by (2), we can reach rung $k + 1$.



Hence, $P(k) \rightarrow P(k + 1)$ is true for all positive integers k .
We can reach every rung on the ladder.

Remembering How Mathematical Induction Works

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

Let $P(n)$ be the proposition that the n th 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 k th 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 .

Examples of Proof by Mathematical Induction

PROVING SUMMATION FORMULAE

Example 1: Show that if n is a positive integer, then $1 + 2 + \dots + n = \frac{n(n+1)}{2}$

Solution: Let $P(n)$ be the proposition that $1 + 2 + \dots + n = \frac{n(n+1)}{2}$.

BASIS STEP: $P(1)$ is true, because $1 = \frac{1(1+1)}{2}$

$$1 = 1$$

INDUCTIVE STEP: Inductive hypothesis that $P(k)$ is true for an arbitrary positive integer k .
The inductive hypothesis $P(k)$ is

$$1 + 2 + \dots + k = \frac{k(k+1)}{2} \text{ for all positive integers } k.$$

Prove that if $P(k)$ is true, then $P(k+1)$, which is the statement that

$$1 + 2 + 3 + \dots + k + (k+1) = \frac{(k+1)[(k+1)+1]}{2} = \frac{(k+1)(k+2)}{2}$$

is also true.

Continue

PROVING SUMMATION FORMULAE

$$1 + 2 + \dots + k \stackrel{\text{IH}}{=} \frac{k(k+1)}{2}$$

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

Add $k+1$ to both sides of the equation

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

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

Hence proved.

Therefore, because we have completed both the basis step and the inductive step, by the principle of mathematical induction we have shown that $1 + 2 + \dots + n = \frac{n(n+1)}{2}$ is true for all positive integers n .

PROVING SUMMATION FORMULAE

Example 2: Conjecture a formula for the sum of the first n positive odd integers. Then prove your conjecture using mathematical induction.

Solution: Let $P(n)$ be the proposition that $1 + 3 + 5 + \dots + (2n - 1) = n^2$ for n positive odd integers

BASIS STEP: $P(1)$ is true, because $1 = 1^2$
 $1 = 1$

INDUCTIVE STEP: Inductive hypothesis that $P(k)$ is true for an arbitrary positive integer k .
The inductive hypothesis $P(k)$ is

$$1 + 3 + 5 + \dots + (2k - 1) = k^2 \text{ for all positive integers } k.$$

Prove that if $P(k)$ is true, then $P(k + 1)$, which is the statement that

$$1 + 3 + 5 + \dots + (2k - 1) + [2(k+1) - 1] = (k + 1)^2$$

$$1 + 3 + 5 + \dots + (2k - 1) + [2k + 2 - 1] = (k + 1)^2$$

$$1 + 3 + 5 + \dots + (2k - 1) + (2k+1) = (k + 1)^2 \text{ is also true.}$$

Continue

PROVING SUMMATION FORMULAE

$$1 + 3 + 5 + \dots + (2k - 1) + (2k+1) = [1 + 3 + 5 + \dots + (2k - 1)] + (2k+1)$$

$$\begin{aligned} & \stackrel{\text{IH}}{=} k^2 + (2k+1) && \text{(by inductive hypothesis)} \\ &= k^2 + 2k + 1 \\ &= (k + 1)^2 \end{aligned}$$

Hence proved.

Therefore, because we have completed both the basis step and the inductive step, by the principle of mathematical induction we have shown that $1 + 3 + 5 + \dots + (2n - 1) = n^2$ is true for all positive integers n .

End

PROVING INEQUALITIES

Example 1: Use mathematical induction to prove the inequality $n < 2^n$ for all positive integers n .

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

BASIS STEP: $P(1)$ is true, because $1 < 2^1$

INDUCTIVE STEP: Inductive hypothesis that $P(k)$ is true for an arbitrary positive integer k . ($k \geq 1$)
The inductive hypothesis $P(k)$ is $k < 2^k$ for all positive integers k .

Prove that if $P(k)$ is true, then $P(k + 1)$, which is the statement that $k + 1 < 2^{k+1}$ is true.

$$\overset{\text{IH}}{k} < 2^k$$

$$k + 1 < 2^k + 1 \quad (\text{Add 1 to both sides})$$

$$\leq 2^k + k \quad (k = 1)$$

$$< 2^k + 2^k \quad (\text{by the inductive hypothesis } k < 2^k)$$

$$< 2 \cdot 2^k \quad (\text{addition})$$

$$k + 1 < 2^{k+1} \quad (\text{Multiplication})$$

Hence proved.

Therefore, because we have completed both the basis step and the inductive step, by the principle of mathematical induction we have shown that $n < 2^n$ is true for all positive integers n .

PROVING INEQUALITIES

Example 2: Use mathematical induction to prove that $2^n < n!$ for every integer n with $n \geq 4$.

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

BASIS STEP: $P(4)$ is true, because $2^4 < 4!$
 $16 < 24$

INDUCTIVE STEP: Inductive hypothesis that $P(k)$ is true for an arbitrary integer k with $k \geq 4$.
The inductive hypothesis $P(k)$ is $2^k < k!$ for the positive integer k with $k \geq 4$.

Prove that if $P(k)$ is true, then $P(k + 1)$, which is the statement that $2^{k+1} < (k + 1)!$ is true.

$$\begin{aligned} 2^{k+1} &= 2 \cdot 2^k && \text{by definition of exponent} \\ &\stackrel{\text{IH}}{<} 2 \cdot k! && \text{by the inductive hypothesis} \\ &< (k + 1)k! && \text{because } 2 < k + 1 \\ 2^{k+1} &< (k + 1)! && \text{by definition of factorial function} \end{aligned}$$

Hence proved.

Therefore, because we have completed both the basis step and the inductive step, by the principle of mathematical induction we have shown that $2^n < n!$ for every integer n with $n \geq 4$.

PROVING DIVISIBILITY RESULTS

Example: Use mathematical induction to prove that $n^3 - n$ is divisible by 3 whenever n is a positive integer.

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

BASIS STEP: $P(1)$ is true, because $1^3 - 1 = 0$ is divisible by 3

INDUCTIVE STEP: Inductive hypothesis that $P(k)$ is true for an arbitrary integer k .

The inductive hypothesis $P(k)$ is $k^3 - k$ is divisible by 3 for the positive integer k .

Prove that if $P(k)$ is true, then $P(k + 1)$, which is the statement that $(k + 1)^3 - (k + 1)$ is divisible by 3 is true.

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

Using the inductive hypothesis, we conclude that the first term $k^3 - k$ is divisible by 3.

The second term is divisible by 3 because it is 3 times an integer.

Hence proved.

Therefore, because we have completed both the basis step and the inductive step, by the principle of mathematical induction we have shown that $n^3 - n$ is divisible by 3 whenever n is a positive integer.

Guidelines for Proofs by Mathematical Induction

Guidelines: Mathematical Induction Proofs



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 the words “**Basis Step**.” Then show that $P(b)$ is true, taking care that the correct value of b is used. This completes the first part of the proof.
3. Write out the words “**Inductive Step**”.
4. State, and clearly identify, the inductive hypothesis, in the form “**assume that $P(k)$ is true for an arbitrary fixed integer $k \geq b$.**”
5. State what needs to be proved under the assumption that the inductive hypothesis is true. That is, write out what $P(k + 1)$ says.
6. **Prove the statement $P(k + 1)$** making use 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 $k = b$.
7. Clearly identify the **conclusion** of the inductive step, such as by saying “this completes the inductive step.”
8. After completing the basis step and the inductive step, state the conclusion, namely, by mathematical induction, **$P(n)$ is true for all integers n with $n \geq b$.**

Review Questions

Let $P(n)$ be the statement that $1^2 + 2^2 + \cdots + n^2 = \frac{n(n+1)(2n+1)}{6}$ 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.
- (f) Explain why these steps show that this formula is true whenever n is a positive integer.

Review Questions

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

(a) What is the statement $P(1)$? a) $1^2 = 1 \cdot 2 \cdot 3 / 6$

(b) Show that $P(1)$ is true, completing the basis step of the proof. b) Both sides of $P(1)$ shown in part (a) equal 1.

(c) What is the inductive hypothesis? c) $1^2 + 2^2 + \cdots + k^2 = k(k+1)(2k+1)/6$

(d) What do you need to prove in the inductive step? d) For each $k \geq 1$ that $P(k)$ implies $P(k+1)$ (OR) we can show
 $1^2 + 2^2 + \cdots + k^2 + (k+1)^2 = (k+1)(k+2)(2k+3)/6$

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

(f) Explain why these steps show that this formula is true whenever n is a positive integer.

f. We have completed both the basis step and the inductive step, so by the principle of mathematical induction, the statement is true for every positive integer n .

Review Questions

$$\begin{aligned} \text{e. } (1^2+2^2+\dots+k^2) + (k+1)^2 &= \frac{[k(k+1)(2k+1)]}{6} + (k+1)^2 \\ &= \frac{[(k+1)] [k(2k+1) + 6(k+1)]}{6} \\ &= \frac{[(k+1)](2k^2 + k + 6k + 1)}{6} \\ &= \frac{[(k+1)](2k^2 + 7k + 6)}{6} \\ &= \frac{[(k+1)](k+2)(2k+3)}{6} \\ &= \frac{(k+1)(k+2)(2k+3)}{6} \end{aligned}$$

f. We have completed both the basis step and the inductive step, so by the principle of mathematical induction, the statement is true for every positive integer n .

Strong Induction

Strong Induction

- **Strong induction** is used when we cannot easily prove a result using mathematical induction.
- The basis step of a proof by strong induction is the same as a proof of the same result using mathematical induction. However, the inductive steps in these two proof methods are different.

PRINCIPLE OF STRONG INDUCTION 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 the proposition $P(1)$ is true.

INDUCTIVE STEP: We show that the conditional statement

$[P(1) \wedge P(2) \wedge \cdots \wedge P(k)] \rightarrow P(k + 1)$ is true for all positive integers k .

The inductive hypothesis is $P(j)$ is true for $j = 1, 2, \dots, k$.

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

Example Proofs using Strong Induction

Completion of the 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 STEP: The inductive hypothesis is $P(j)$ is true for all integers j with $2 \leq j \leq k$. To 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, it has been shown that every integer greater than 1 can be written as the product of primes.

Proof using Strong Induction₂



- **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 STEP:** The inductive hypothesis states that $P(j)$ holds for $12 \leq j \leq k$, where $k \geq 15$. Assuming the inductive hypothesis, it can be shown 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 $n \geq 12$.

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 STEP: The inductive hypothesis $P(k)$ for any positive integer k is that postage of k cents can be formed using 4-cent and 5-cent stamps. To 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 $n \geq 12$.

Thank you





Discrete Structures

(CKC111)

Week 11 & Week 12



Induction and Recursion

Recursive Definitions and Structural Induction

Section Summary



- ✓ Recursively Defined Functions
- ✓ Recursively Defined Sets and Structures

Introduction

- The object is defined in terms of itself. This process is called recursion.
- We can use recursion to define sequences, functions, and sets.

Example:

The sequence of powers of 2 is given by $a_n = 2^n$ for $n = 0, 1, 2, \dots$

Recursively Defined Set

BASIS STEP:

$$a_0 = 1$$

RECURSIVE STEP:

$$a_{n+1} = 2a_n \text{ for } n = 0, 1, 2, \dots$$

The terms of the sequence are found from previous terms.

- To prove results about recursively defined sets we use a method called *structural induction*.
- Structural induction: a technique for proving results about recursively defined sets

Recursively Defined Functions

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, \dots , where a_i is a real number for every nonnegative integer i , where $f(i) = a_i$. *This was done using recurrence relations in Section 2.4 (main reference book).*

Recursive definition of a function: a definition of a function that specifies an initial set of values and a rule for obtaining values of this function at integers from its values at smaller integers.

Recursively Defined Functions

Example 1:

Suppose that f is defined recursively by

BASIS STEP: $f(0)=3$

RECURSIVE STEP: $f(n+1)=2f(n)+3$

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

Solution: From the recursive definition it follows that

$$f(1)=2f(0)+3=2\cdot 3+3=9,$$

$$f(2)=2f(1)+3=2\cdot 9+3=21,$$

$$f(3)=2f(2)+3=2\cdot 21+3=45,$$

$$f(4)=2f(3)+3=2\cdot 45+3=93.$$

Recursively Defined Functions

Student
Work

Example 2:

Suppose that f is defined recursively by

BASIS STEP: $f_0 = 0, f_1 = 1$

RECURSIVE STEP: $f_n = f_{n-1} + f_{n-2}$

Find the Fibonacci numbers, f_2, f_3, f_4 and f_5

Solution: From the recursive definition it follows that

$$f_2 = f_1 + f_0 = 0 + 1 = 1$$

$$f_3 = f_2 + f_1 = 1 + 1 = 2$$

$$f_4 = f_3 + f_2 = 2 + 1 = 3$$

$$f_5 = f_4 + f_3 = 3 + 2 = 5$$

Recursively Defined Functions

Solution:

$$f(n) = a^n$$

BASIS STEP: $f(0) = a^0 = 1$

RECURSIVE STEP:

Find the rule for $f(n+1)$ from $f(n)$,

$$f(1) = a^1 = a$$

$$f(2) = a^2 = a \cdot a = a \cdot a^1$$

$$f(3) = a^3 = a \cdot a \cdot a = a \cdot a^2$$

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

⋮

$$f(n) = a^n = a \cdot a \cdot \dots \cdot a = a \cdot a^{n-1}$$

$$f(n+1) = a^{n+1} = a \cdot a^n$$

$$f(n+1) = a \cdot f(n) \text{ for } n = 0, 1, 2, 3, \dots$$

Example 3:

Give a recursive definition of a^n , where a is a nonzero real number and n is a nonnegative integer

Recursively Defined Functions

Example 4:

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

Solution:

$$f(n) = n!$$

BASIS STEP: $f(0) = 0! = 1$

RECURSIVE STEP:

Find the rule for $f(n+1)$ from $f(n)$,

$$f(1) = 1,$$

$$f(2) = (2) \cdot 1 = 2,$$

$$f(3) = (3) \cdot 2 = 6,$$

$$f(4) = (4) \cdot 6 = 24.$$

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

for $n = 0, 1, 2, 3, \dots$

$$f(0) = 0! = 1$$

$$f(1) = 1! = 1$$

$$f(2) = 2! = 2 \times 1$$

$$f(3) = 3! = 3 \times 2 \times 1$$

$$f(4) = 4! = 4 \times 3 \times 2 \times 1$$

.

.

$$f(n) = n! = n \times \dots \times 4 \times 3 \times 2 \times 1$$

$$= 2 \times 1!$$

$$= 3 \times 2!$$

$$= 4 \times 3!$$

.

.

$$= n \times (n-1)!$$

$$f(n+1) = (n+1)! = (n+1) \times n \times \dots \times 4 \times 3 \times 2 \times 1 = (n+1) \times n! = (n+1) \cdot f(n)$$

Recursively Defined Functions



Example 3:

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

Student
Work

Recursively Defined Sets and Structures

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 (is true), even if it is not explicitly/clearly mentioned.

Recursive definition of a set: a definition of a set that specifies an initial set of elements in the set and a rule for obtaining other elements from those in the set.

Recursively Defined Sets and Structures

<https://www.youtube.com/watch?v=WstKQxUYgnY>

Example 1:

Recursive definition of a set S containing the positive multiples of 3.

- Basis Step: $3 \in S$
- Recursive Step: If $n \in S$ then $n + 3 \in S$

Compute the elements in the set S .

$$S = \{ 3, 6, 9, 12, \dots \}$$

Example 2:

What elements are in the set T defined by:

- Basis Step: $1 \in T$
- Recursive Step: If $n \in T$ then $n+1 \in T$ and $n-1 \in T$

$$T = \{ 1, 2, 0, 3, -1, 4, -2, \dots \}$$

$$T = \{ \dots, -2, -1, 0, 1, 2, 3, \dots \}$$

Student
Work

Recursively Defined Sets and Structures



Student
Work

Example 1: Subset of Integers S :

Basis step: $3 \in S$.

Recursive step: If $x \in S$ and $y \in S$, then $x + y$ is in S .

Initially 3 is in S , then $3 + 3 = 6$, then $3 + 6 = 9$, etc.

Example 2: The natural numbers N .

Basis step: $0 \in N$.

Recursive step: If n is in N , then $n + 1$ is in N .

Initially 0 is in S , then $0 + 1 = 1$, then $1 + 1 = 2$, etc.

Recursively Defined Sets and Structures

<https://www.youtube.com/watch?v=WstKQxUYgnY>

Write out a recursive definition of a set containing the powers of 3 (starting at 1)
 $\{1, 3, 9, 27, 81, \dots\}$

Solution:

BASIS STEP: $1 \in S$

RECURSIVE STEP: If $n \in S$, then $3n \in S$

Student
Work

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, \dots, T_n are disjoint rooted trees with roots r_1, r_2, \dots, 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, \dots, T_n , and adding an edge from r to each of the vertices r_1, r_2, \dots, r_n , is also a rooted tree.

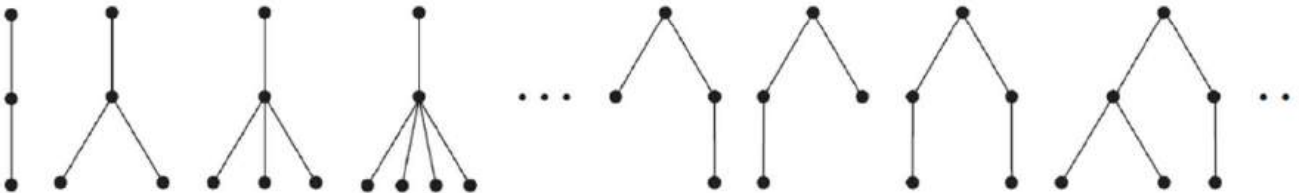
Basis step



Step 1



Step 2



Some of rooted trees formed starting with the basis step and applying the recursive step one time and two times. Note that infinitely many rooted trees are formed at each application of the recursive definition.

Building Up Extended Binary Trees

Binary trees

- special type of rooted trees
- two binary trees are combined to form a new tree with one of these trees designated the left subtree and the other the right subtree.

In extended binary trees, the left subtree or the right subtree can be empty, but in full binary trees this is not possible. Binary trees are one of the most important types of structures in computer science.

Basis step: The empty set is an extended binary tree.

Recursive step: If T_1 and T_2 are disjoint extended binary trees, there is an extended 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 when these trees are nonempty.

Basis step \emptyset

Step 1 \bullet



Step 3

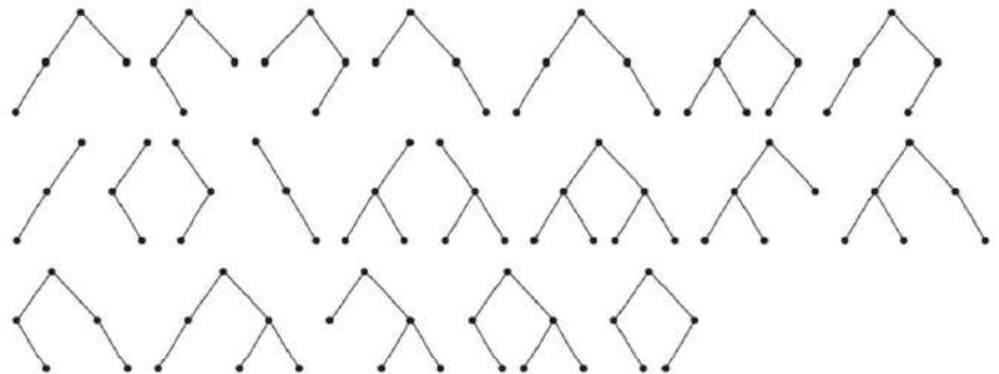


Figure shows how extended binary trees are built up by applying the recursive step from one to three times.

Building Up Full Binary Rooted Trees

Definition: Note that the difference between this recursive definition and that of extended binary trees lies entirely in the basis step.

Basis step: There is a full binary tree consisting only of a single vertex r .

Recursive step : If T_1 and T_2 are disjoint full binary trees, there is 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 .

Basis step



Step 1



Step 2



Figure shows how full binary trees are built up by applying the recursive step one to two times.

Structural Induction

Structural Induction



To prove results about recursively defined sets, we generally use some form of mathematical induction. This example illustrates the connection between recursively defined sets and mathematical induction.

Example 1: Show that the set S by specifying that $3 \in S$ and that if $x \in S$ and $y \in S$, then $x + y \in S$, is the set of all **positive integers** that are **multiples of 3**.

Solution: Let A be the set of all positive integers divisible by 3. To prove that $A = S$, show that A is a subset of S and S is a subset of A . To prove that A is a subset of S , we must show that every positive integer divisible by 3 is in S .

Mathematical induction

$A \subset S$:

Let $P(n)$ be the statement that $3n$ belongs to S .

Basis step: $3 \cdot 1 = 3 \in S$, by the first part of recursive definition.

Inductive step: Assume $P(k)$ is true. By the second part of the recursive definition, if $3k \in S$, then since $3 \in S$, $3k + 3 = 3(k + 1) \in S$. Hence, $P(k + 1)$ is true.

Recursively defined sets

$S \subset A$:

Basis step: $3 \in S$ by the first part of recursive definition, and $3 = 3 \cdot 1$.

Recursive step: The second part of the recursive definition adds $x + y$ to S , if both x and y are in S . If x and y are both in A , then both x and y are divisible by 3.

Structural Induction

We used mathematical induction over the set of positive integers and a recursive definition to prove a result about a recursively defined set. However, instead of using mathematical induction directly to prove results about recursively defined sets, we can use a more convenient form of induction known as **structural induction**. A proof by structural induction consists of two parts. These parts are

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

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.

let $P(n)$ state that the claim is true for all elements of the set that are generated by n or fewer applications of the rules in the recursive step of a recursive definition.

Structural Induction

Let S be the subset of the set of integers defined recursively by:

Basis Step: $3 \in S$

Recursive Step: If $a \in S$ and $b \in S$, then $a + b \in S$

Use structural induction to show that $3|x$ for all $x \in S$.

$$\begin{aligned} 3|x \\ x = 3(m) \end{aligned}$$

Let $P(n)$ be that $3|x$ for all $x \in S$ after n applications of the recursive definition

Basis Step: $3 \in S$,

$$3 | 3$$

$$3 = 3(1)$$

Recursive Step:

$$a \in S, \text{ So } a = 3m \quad b \in S, \text{ So } b = 3n$$

$$a + b = 3m + 3n$$

$$a + b = 3(m + n)$$

$$\text{Therefore, } 3|(a + b)$$

Structural Induction

Let S be the subset of the set of ordered pairs of integers defines by:

Basis Step: $(0,0) \in S$

Recursive Step: If $(a, b) \in S$, then $(a, b + 1) \in S$, $(a + 1, b + 1) \in S$, $(a + 2, b + 1) \in S$

Use structural induction to show that $a \leq 2b$ whenever $(a, b) \in S$

Let $P(n)$ be that $a \leq 2b$ whenever $(a, b) \in S$ after n applications of the recursive definition

Basis Step: $(0,0) \in S$,

$$0 \leq 2(0)$$

$$0 \leq 0$$

Recursive Step:

	$(a, b + 1) \in S$	$(a + 1, b + 1) \in S$	$(a + 2, b + 1) \in S$
$a \leq 2b$	$a \leq 2(b + 1)$	$a + 1 \leq 2(b + 1)$	$a + 2 \leq 2(b + 1)$
	$0 \leq 2(0 + 1)$	$0 + 1 \leq 2(0 + 1)$	$0 + 2 \leq 2(0 + 1)$
	$0 \leq 2$	$1 \leq 2$	$2 \leq 2$

Thank you





Discrete Structures

(CKC111)

Week 11 & Week 12



Induction and Recursion

Recursive Algorithms

Section Summary



- ✓ Recursive Algorithms
- ✓ Proving Recursive Algorithms Correct
- ✓ Merge Sort

Recursive Algorithms

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.

Algorithm 1: 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  $n \cdot \text{factorial}(n - 1)$ 
  {output is  $n!$ }
```

Algorithm 1: Recursive Factorial Algorithm

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

To help understand – trace the steps, e.g., compute $4!$.

$$4! = 4 \cdot 3!$$

$$3! = 3 \cdot 2!$$

$$2! = 2 \cdot 1!$$

$$1! = 1 \cdot 0!.$$

Inserting the value of $0! = 1$

Working back through the steps,

$$1! = 1 \cdot 1 = 1$$

$$2! = 2 \cdot 1! = 2 \cdot 1 = 2$$

$$3! = 3 \cdot 2! = 3 \cdot 2 = 6$$

$$4! = 4 \cdot 3! = 4 \cdot 6 = 24.$$

Algorithm 2: 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  $a \cdot \text{power}(a, n - 1)$ 
{output is  $a^n$ }
```

A Recursive Algorithm for Computing a^n .

Student
Work

Trace the steps when the input is $a = 2, n = 5$

$$2^5 = 2 \cdot 2^4$$

$$2^4 = 2 \cdot 2^3$$

$$2^3 = 2 \cdot 2^2$$

$$2^2 = 2 \cdot 2^1$$

$$2^1 = 2 \cdot 2^0$$

Inserting the value of $2^0 = 1$

Working back through the steps,

$$2^1 = 2 \cdot 1 = 2$$

$$2^2 = 2 \cdot 2 = 4$$

$$2^3 = 2 \cdot 4 = 8$$

$$2^4 = 2 \cdot 8 = 16$$

$$2^5 = 2 \cdot 16 = 32$$

Algorithm 3: Recursive GCD Algorithm



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

Solution: Use the reduction $\text{gcd}(a, b) = \text{gcd}(b \bmod a, a)$ and the condition $\text{gcd}(0, b) = b$ when $b > 0$.

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

A Recursive Algorithm for Computing $\gcd(a, b)$.

Student
Work

Trace the steps when the input is $a = 5, b = 8$.

$$\gcd(5, 8) = \gcd(8 \bmod 5, 5) = \gcd(3, 5).$$

$$\gcd(3, 5) = \gcd(5 \bmod 3, 3) = \gcd(2, 3),$$

$$\gcd(2, 3) = \gcd(3 \bmod 2, 2) = \gcd(1, 2),$$

$$\gcd(1, 2) = \gcd(2 \bmod 1, 1) = \gcd(0, 1).$$

$\gcd(0, 1)$ it uses the first step with $a = 0$ to find that $\gcd(0, 1) = 1$.

Consequently, the algorithm finds that $\gcd(5, 8) = 1$.

Algorithm 4: Recursive Linear Search Algorithm



Example: Express the linear search algorithm as a recursive procedure.

Solution: To search for the first occurrence, of x in the sequence a_1, a_2, \dots, a_n , at the i^{th} step of the algorithm, x and a_i are compared. If equals, returns i . Otherwise, the search for the first occurrence of x is reduced to a search in a sequence with one fewer element, namely, the sequence a_{i+1}, \dots, a_n . When x never found, returns 0.

```
procedure search( $i, j, x$ :  $i, j, x$  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  $a_1, a_2, \dots, a_n$  if it appears; otherwise,  
it is 0}
```

Algorithm 5: Recursive Binary Search Algorithm



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

Solution: To locate x in the sequence a_1, a_2, \dots, a_n , in increasing order. Begin by comparing x with the middle term, $a_{\lfloor (n+1)/2 \rfloor}$. If equals, algorithm will terminate and return the location of this term. Otherwise, it reduce to a search the search to a smaller search sequence.

```
procedure binary search( $i, j, x$ :  $i, j, x$  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 the location of  $x$  in  $a_1, a_2, \dots, a_n$  if it appears; otherwise, it is 0}
```

Proving Recursive Algorithms Correct

Proving Recursive Algorithms Correct



Both mathematical and strong induction are useful techniques to show that recursive algorithms always produce the correct output.

Example: Prove that the algorithm for computing the powers of real numbers is correct.

```
procedure power(a: nonzero real number, n: nonnegative integer)
  if n = 0 then return 1
  else return a · power(a, n − 1)
  {output is  $a^n$ }
```

Solution: Use mathematical induction on the exponent n .

Basis step: If $n=0$, $power(a, 0) = 1$. This is correct because $a^0 = 1$ for every nonzero real number a .

Inductive step: The inductive hypothesis is that $power(a, k) = a^k$, for all $a \neq 0$. the inductive hypothesis is the algorithm correctly computes a^k . Next, we prove the inductive hypothesis is true, then the algorithm correctly computes a^{k+1} , since $power(a, k + 1) = a \cdot power(a, k) = a \cdot a^k = a^{k+1}$.

Proving Recursive Algorithms Correct



Both mathematical and strong induction are useful techniques to show that recursive algorithms always produce the correct output.

Example: Prove that the algorithm for computing the powers of real numbers is correct.

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

Solution: Use mathematical induction on the exponent n .

Basis step: If $n=0$, $\text{power}(a, 0) = 1$. This is correct because

Inductive step: The inductive hypothesis is that $\text{power}(a, k)$ hypothesis is the algorithm correctly computes a^k . Next, then the algorithm correctly computes a^{k+1} , since $\text{power}(a, k + 1) = a \cdot \text{power}(a, k) = a \cdot a^k = a^{k+1}$.

We have completed the basis step and the inductive step, so we can conclude that this algorithm always computes a_n correctly when $a \neq 0$ and n is a nonnegative integer.

Merge Sort

Merge Sort



Merge Sort works by iteratively splitting a list (with an even number of elements) into two sub-lists of equal length until each sublist has one element.

Each sublist is represented by a balanced binary tree.

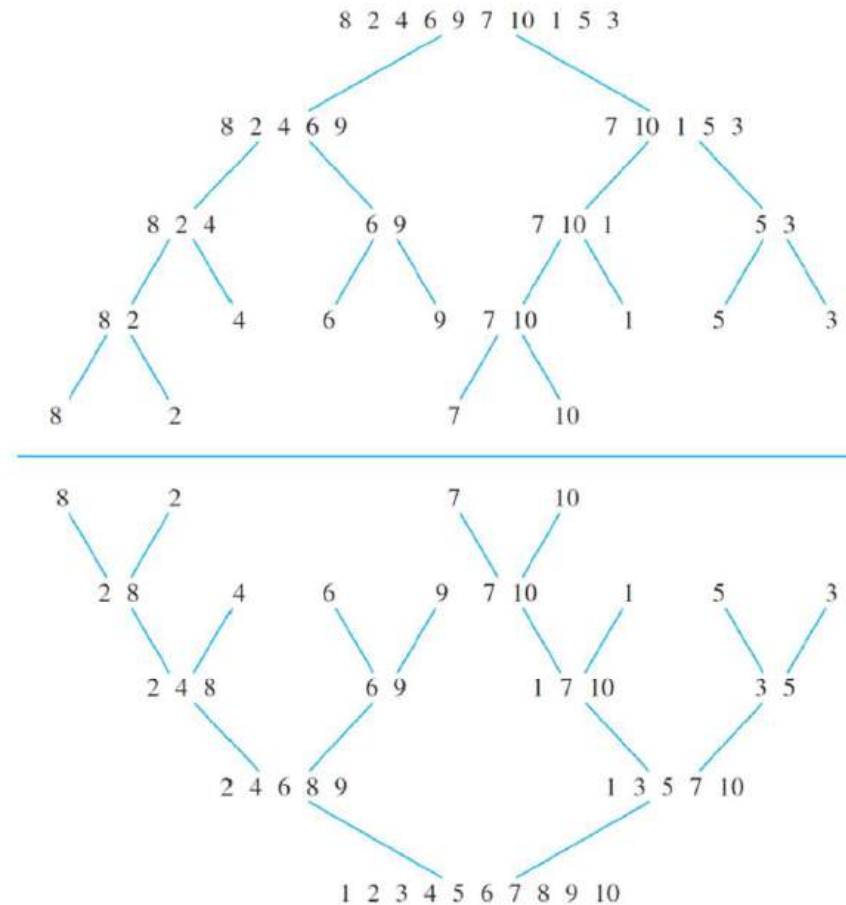
At each step a pair of sublists is successively merged into a list with the elements in increasing order. The process ends when all the sublists have been merged.

The succession of merged lists is represented by a binary tree.

Merge Sort

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

Solution:



Recursive Merge Sort

Example: Construct a recursive merge sort algorithm.

Solution: Begin with the list of n elements L .

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

Merging Two List

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	

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}

Thank you

