Chapter 5.

Induction and recursion

歸納 遞迴

5.1 Mathematical Induction 數学歸納法

強數学歸納法 良序

5.2 Strong Induction and Well-Ordering

5.3 Recursive Definitions and Structural Induction

5.4 Recursive Algorithms 遞迴算法

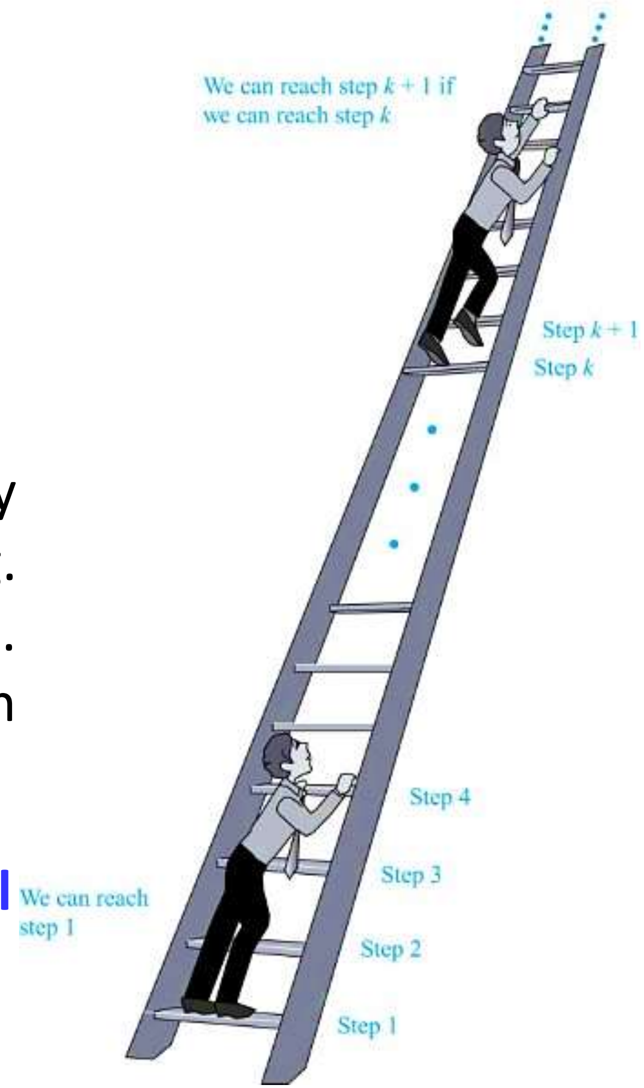
Climbing an Infinite Ladder

Suppose we have an infinite ladder:

1. We can reach the first rung of the ladder.
   階
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

Principle of Mathematical Induction (數學歸納法)

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 Step***: **Show that $P(k) \rightarrow P(k + 1)$ 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 must $P(k + 1)$ be true.**

〈利用數学歸納法〉
①證明n=1是成立的
②假设n=k成立，證明n=k+1成立。

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.

Important Points About Using Mathematical Induction

Mathematical induction can be expressed as the rule of inference

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

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, 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.

Validity of Mathematical Induction

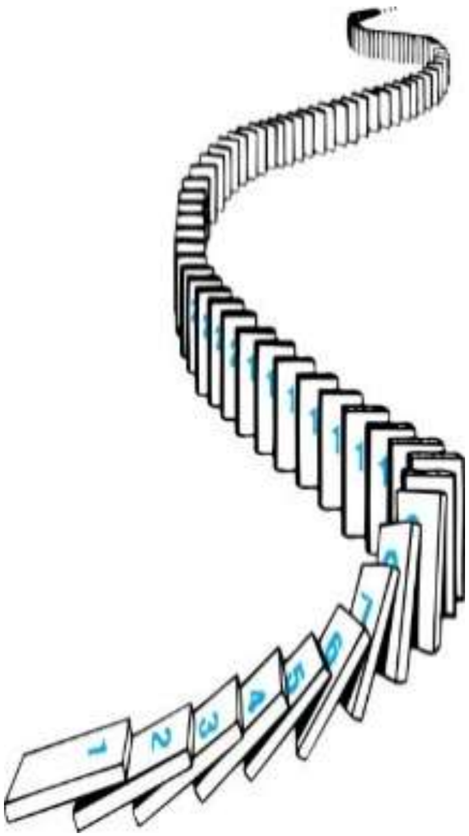
Mathematical induction is valid because of the well ordering property, which states that every nonempty subset of the set of positive integers has a least element (*see Section* 5.2 *and Appendix* 1). Here is the proof:

- 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.

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

Proving a Summation Formula by Mathematical Induction

Example: Show that: 求和公式 $$\boxed{\sum_{i=1}^{n} i = \frac{n(n+1)}{2}}$$

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

Solution:

- BASIS STEP: $P(1)$ is true since $1(1 + 1)/2 = 1$. $\sum_{i=1}^{1} i = 1$
- INDUCTIVE STEP: Assume true for $P(k)$.

利用數学歸納法：
① 證明n=1是成立的
②假设n=k成立，
證明n=k+1成立！

The inductive hypothesis is ② 假设 $$\sum_{i=1}^{k} i = \frac{k(k+1)}{2}$$

Under this assumption,

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

② 證明n=k+1 成立

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

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

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,

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

when $n=k+1$

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

利用數学歸納法：

① 證明 $n=1$ 成立

② 假設 $n=k$ 成立

證明 $n=k+1$ 成立

- We prove the conjecture is proved correct with mathematical induction.
- BASIS STEP: *P*(1) is true since $1^2 = 1$.
- INDUCTIVE STEP: *P(k)* → *P*(*k* + 1) for every positive integer *k*.

  Assume the inductive hypothesis holds and then show that *P*(*k* + 1) holds has well.

Inductive Hypothesis: $1+3+5+\cdots+(2k-1)=k^2$

- So, assuming *P*(*k*), it follows that:

$$\begin{aligned}1+3+5+\cdots+(2k-1)+(2k+1) &= \left[1+3+5+\cdots+(2k-1)\right]+(2k+1)\\ &= k^2+(2k+1)\ (\textit{by the inductive hypothesis})\\ &= k^2+2k+1\\ &= (k+1)^2\end{aligned}$$

- Hence, we have shown that *P*(*k* + 1) follows from *P*(*k*). Therefore the sum of the first *n* positive odd integers is n^2.

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 STEP: $P(1)$ is true since $1 < 2^1 = 2$. ← 證明 n=1 成立
- INDUCTIVE STEP: Assume $P(k)$ holds, i.e., $k < 2^k$, for an arbitrary positive integer k. ← 假设 n=k 是成立的！
- Must show that $P(k + 1)$ holds. Since by the inductive hypothesis, $k < 2^k$, it follows that: ← 證明 n=k+1 是成立的

$$k + 1 < 2^k + 1 \leq 2^k + 2^k = 2 \cdot 2^k = 2^{k+1}$$

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

Proving Inequalities₂

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

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

- BASIS STEP: $P(4)$ is true since $2^4 = 16 < 4! = 24$.
- INDUCTIVE STEP: Assume $P(k)$ holds, i.e., $2^k < k!$ for an arbitrary integer $k \geq 4$. To show that $P(k + 1)$ holds:

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

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

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

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 STEP: $P(1)$ is true since $1^3 - 1 = 0$, which is divisible by 3.
- INDUCTIVE STEP: Assume $P(k)$ holds, i.e., $k^3 - k$ is divisible by 3, for an arbitrary positive integer k. To show that $P(k + 1)$ follows:

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

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.

Number of Subsets of a Finite Set[1]

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**.

從 0 開始!

(*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 Step: ***P*(0) is true**, because the empty set has only itself as a subset and $2^0 = 1$.
- Inductive Step: Assume $P(k)$ is true for an arbitrary nonnegative integer k.

Number of Subsets of a Finite Set₂

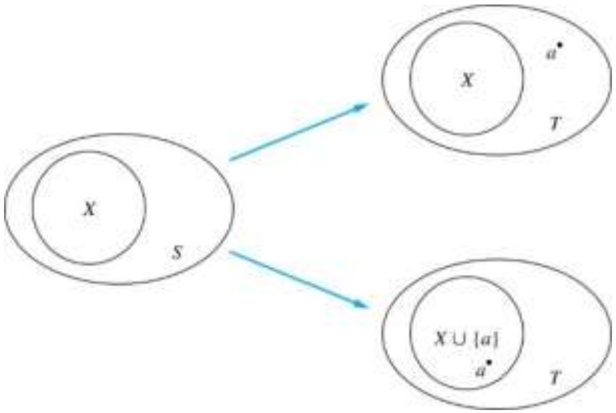
任意的

Inductive Hypothesis: For an arbitrary nonnegative integer k, every set with k elements has 2^k subsets.

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

$S = \{\underbrace{x_1, x_2, x_3, \ldots\ldots x_k}_{k\text{個}}, a\}$

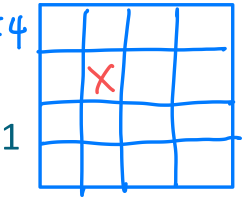
For each subset X of S, there are exactly two subsets of T, i.e., X and $X \cup \{a\}$.



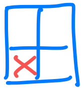
By the inductive hypothesis S has 2^k subsets. Since there are two subsets of T for each subset of S, the number of subsets of T is

$$2 \cdot 2^k = 2^{k+1}.$$

Tiling Checkerboards[1]


4x4

→2x2、4x4……


2x2

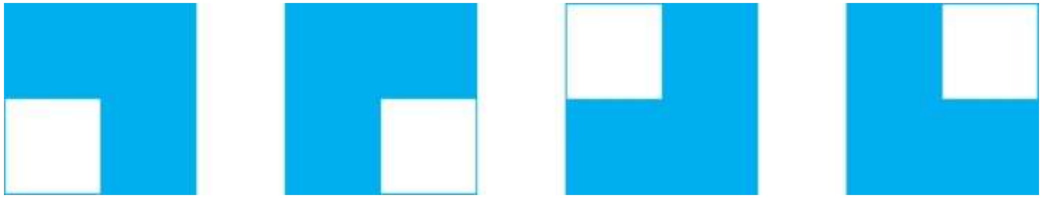
，把任意一個隔子移掉

Example: Show that every $2^n \times 2^n$ checkerboard with one square removed **can be tiled using right triominoes.** →都可以用右側的磁磚來填滿

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



- INDUCTIVE STEP: Assume that $P(k)$ is true for every $2^k \times 2^k$ checkerboard, for some positive integer k.

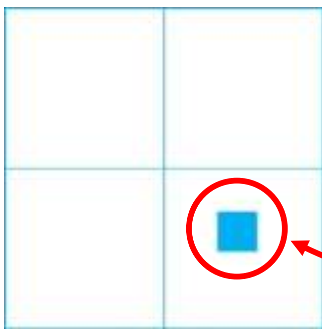
Jump to long description

Tiling Checkerboards₂

假设

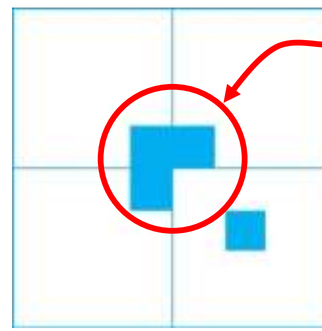
Inductive Hypothesis: Every $2^k \times 2^k$ checkerboard, for some positive integer k, with one square removed can be tiled using right triominoes.

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.



把一個$2^{k+1} \times 2^{k+1}$
分成4個$2^k \times 2^k$區域

假設有一個被移除
掉的區域在這裡



再用一個right triomino
鋪在此位置上

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

An Incorrect "Proof" by Mathematical Induction[1]

Example: Let $P(n)$ be the statement that every set of n lines in the plane, no two of which are parallel, meet in a common point. Here is a "proof" that $P(n)$ is true for all positive integers $n \geq 2$.

?: ✓ 正確

- BASIS STEP: **The statement $P(2)$ is true** because any two lines in the plane that are not parallel meet in a common point.

- INDUCTIVE STEP: The inductive hypothesis is the statement that ***P*(*k*) is true for the positive integer $k \geq 2$**, i.e., every set of k lines in the plane, no two of which are parallel, meet in a common point.

?: ✓ 假设，没有对错

- We must show that if $P(k)$ holds, then $P(k + 1)$ holds, i.e., if every set of k lines in the plane, no two of which are parallel, $k \geq 2$, meet in a common point, then every set of k + 1 lines in the plane, no two of which are parallel, meet in a common point.

An Incorrect "Proof" by Mathematical Induction[2]

X錯了!

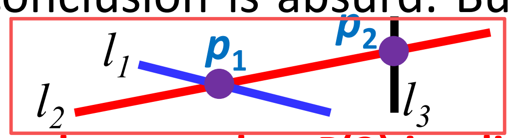
?

Inductive Hypothesis: Every set of k lines in the plane, where $k \geq 2$, no two of which are parallel, meet in a common point.

Consider a set of $k + 1$ distinct lines in the plane, no two parallel. By the inductive hypothesis, **the first k of these lines must meet in a common point p_1**. By the inductive hypothesis, **the last k of these lines meet in a common point p_2**.

If p_1 and p_2 are different points, all lines containing both of them must be the same line since two points determine a line. This contradicts the assumption that the lines are distinct. **Hence, $p_1 = p_2$** lies on all $k + 1$ distinct lines, and therefore $P(k + 1)$ holds. Assuming that $k \geq 2$, distinct lines meet in a common point, then every $k + 1$ lines meet in a common point.

There must be an error in this proof since the conclusion is absurd. But **where is the error?**



- **Answer**: $P(k) \to P(k + 1)$ only holds for $k \geq 3$. **It is not the case that $P(2)$ implies $P(3)$**. The first two lines must meet in a common point p_1 and the second two must meet in a common point p_2. They do not have to be the same point since only the second line is common to both sets of lines.

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 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 $n \geq b$**.

Chapter 5.

Induction and recursion

強歸納法 & 良序

5.1 Mathematical Induction

5.2 Strong Induction and Well-Ordering

5.3 Recursive Definitions and Structural Induction

5.4 Recursive Algorithms

Strong Induction (強歸納法)

20 Principle of Mathematical Induction (數學歸納法)

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 Step*: Show that $P(k) \to P(k+1)$ is true for all positive integers k.

"第" k 步

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

- *Basis Step*: Verify that the proposition $P(1)$ is true.
- *Inductive Step*: Show the conditional statement

$$\boxed{\boxed{\left[P(1) \wedge P(2) \wedge \cdots \wedge P(k)\right]} \to P(k+1)}$$

holds for all positive integers k.

"前" k 步

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

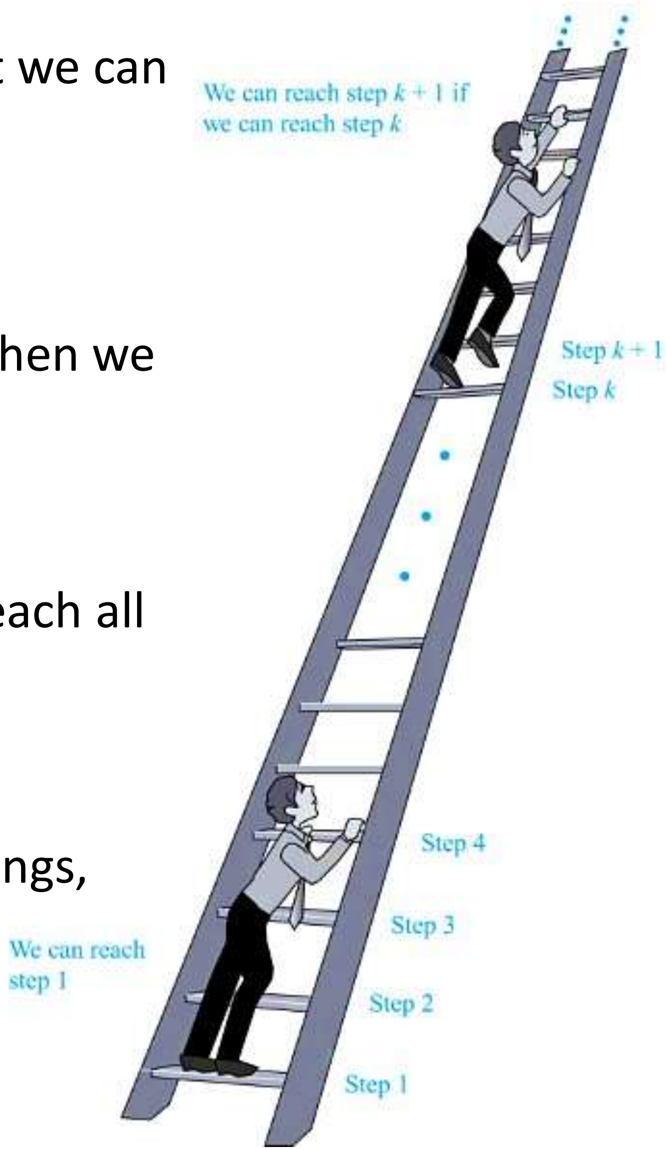
Strong Induction and the Infinite Ladder [1]

Mathematical Induction (數學歸納法) tells us that we can reach all rungs if:

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.

Strong induction (強歸納法) tells us that we can reach all rungs if:

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



Strong Induction and the Infinite Ladder 2

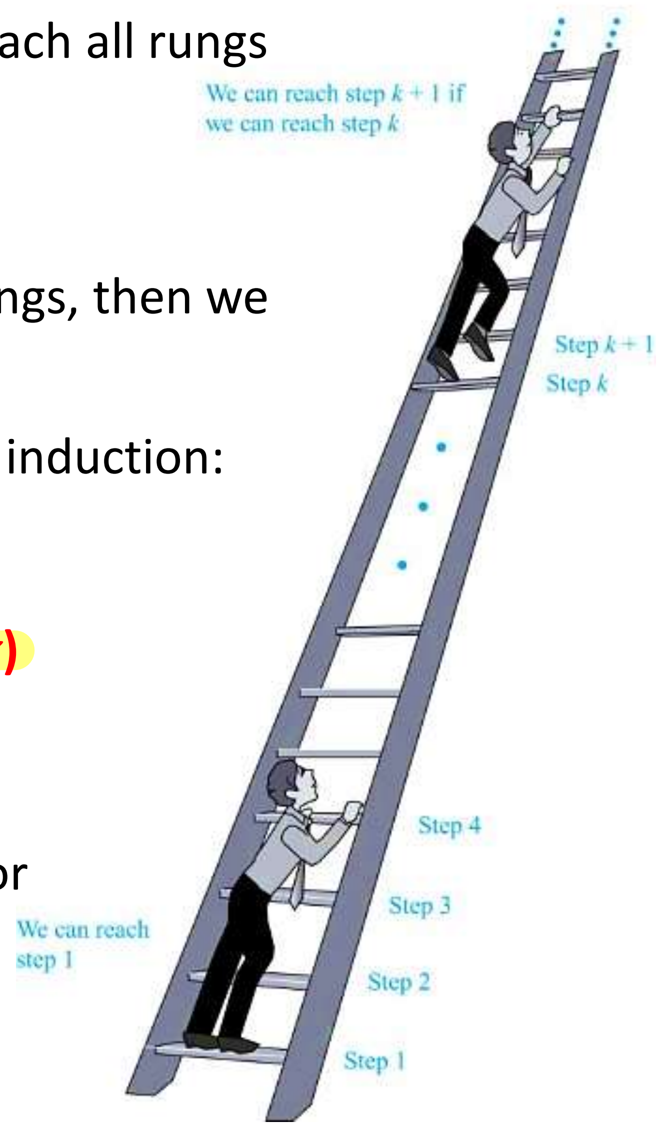
Strong induction (強歸納法) tells us that we can reach all rungs if:

1. We can reach the first rung of the ladder.
2. For every integer k, if we can reach **the first k** rungs, 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 STEP: **Assume $P(1) \wedge P(2) \wedge \cdots \wedge P(k)$ holds** for an arbitrary integer k, and 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.



Proof using Mathematical 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.

Solution: Prove the result **using mathematical induction**.

- BASIS STEP: We can reach the first step. ← According to Example.
- INDUCTIVE STEP: The inductive hypothesis is that **we can reach the kth rungs, for any $k \geq 2$.**
- **However, there is no obvious way to know that we can reach the $(k + 1)$st rung from the kth rung.**

Proof using Strong Induction [1]

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.

Solution: Prove the result **using strong induction.** ← According Problem Description

- BASIS STEP: We can reach the first step.
- INDUCTIVE STEP: The inductive hypothesis is that **we can reach the first k rungs, for any $k \geq 2$.** We can reach the $(k + 1)$st rung since we can reach the $(k - 1)$st rung by the inductive hypothesis.
- Hence, we can reach all rungs of the ladder.

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**. (*See page* 335 *of text*.)

In fact, **the principles of mathematical induction, strong induction, and the well-ordering property are all equivalent.** (*Exercises* 41-43)

Sometimes it is clear how to proceed using one of the three methods, but not the other two.

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.** 可以被質因數分解

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 $2 \leq j \leq k$.** 假设前k項都是成立的 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.

(*uniqueness proved in Section* 4.3)

Proof using Mathematical Induction

$12 = 4+4+4$
$13 = 4+4+5$ ......

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.

$15 = 5+5+5$
-1 -1 -1 +3 +1 下一項
$16 = 4+4+4+4$

Hence, $P(n)$ holds for all $n \geq 12$.

Proof using Strong Induction 2

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 $12 \le j \le k$, where $k \ge 15$.** Assuming the inductive hypothesis, it can be shown that $P(k + 1)$ holds.

when $k=19$ $19-3=16$

- Using the inductive hypothesis, **$P(k - 3)$ holds since $k - 3 \ge 12$**. **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 $n \ge 12$.

$4+16$

Well-Ordering Property[1]

良序 性质

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, as the next example illustrates.

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.

Well-Ordering Property[2]

Example: Use the well-ordering property to prove the division algorithm, which states that if a is an integer and d is a positive integer, then **there are unique integers q and r with $0 \leq r < d$, such that $a = dq + r$.**

Solution: Let S be the set of nonnegative integers of the form $a - dq$, where q is an integer. The set is nonempty since $-dq$ can be made as large as needed.

- By the well-ordering property, **S has a least element $r = a - dq_0$.** The integer **r is nonnegative**. It also must be the case that $r < d$. If it were not (i.e., $r \geq d$), then there would be a smaller nonnegative element in S, namely, $\boxed{a - d(q_0 + 1)} = a - dq_0 - d = r - d \geq 0$.
- Therefore, there are integers q and r with $0 \leq r < d$.

(*uniqueness of q and r is Exercise* 37)

Chapter 5.
Induction and recursion

5.1 Mathematical Induction

5.2 Strong Induction and Well-Ordering

5.3 Recursive Definitions and Structural Induction

5.4 Recursive Algorithms

遞迴定義 &
結構歸納

Recursively Defined Functions[1]

Definition: A ***recursive* (遞迴)** 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.

Recursively Defined Functions 2

Example: Suppose f is defined by:

$$f(0) = 3,$$

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

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

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: $f(0) = 1$

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

Recursively Defined Functions[3]

Example: Give a recursive definition of:

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

Solution: The first part of the definition is

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

The second part is

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

Fibonacci Numbers 1

Fibonacci
(1170- 1250)

Example : The **Fibonacci numbers (費式數列)** are defined as follows:

$$\boxed{\begin{aligned} f_0 &= 0 \\ f_1 &= 1 \\ f_n &= f_{n-1} + f_{n-2} \end{aligned}}$$

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.

Fibonacci Numbers 2

證費式數列第N項必大於 α^{n-2}

Example 4: Show that whenever $n \geq 3$, $f_n > \alpha^{n-2}$, where $\alpha = \left(1+\sqrt{5}\right)/2$.

Solution: Let $P(n)$ be the statement $\boxed{f_n > \alpha^{n-2}.}$

Use strong induction to show that $P(n)$ is true whenever $n \geq 3$.

- BASIS STEP: ***P*(3) holds** since $\alpha < 2 = f_3$

  ***P*(4) holds** since $\alpha^2 = \left(3+\sqrt{5}\right)/2 < 3 = f_4.$
- INDUCTIVE STEP: Assume that $P(j)$ holds, i.e., **$f_j > \alpha^{j-2}$ for all integers j with $3 \leq j \leq k$, where $k \geq 4$.** Show that $P(k + 1)$ holds, i.e., $f_{k+1} > \alpha^{k-1}.$
  - Since $\alpha^2 = \alpha + 1$ (because α is a solution of $x^2 - x - 1 = 0$),

    $$\alpha^{k-1} = \alpha^2 \cdot \alpha^{k-3} = (\alpha+1)\cdot\alpha^{k-3} = \alpha\cdot\alpha^{k-3} + 1\cdot\alpha^{k-3} = \alpha^{k-2} + \alpha^{k-3}$$
  - By the inductive hypothesis, because $k \geq 4$ we have

    $$f_{k-1} > \alpha^{k-3}, \qquad f_k > \alpha^{k-2}.$$
  - Therefore, it follows that

    $$f_{k+1} = f_k + f_{k-1} > \alpha^{k-2} + \alpha^{k-3} = \alpha^{k-1}.$$
  - Hence, $P(k + 1)$ is true.

Why does this equality hold?

Lamé's Theorem [1]


Gabriel Lamé
(1795-1870)

拉梅定理

Lamé's Theorem: Let a and b be positive integers with $a \geq b$. Then the number of divisions used by the Euclidian algorithm to find gcd(a,b) is **less than or equal to five times** the **number of decimal digits in b**.

Proof: When we use the Euclidian algorithm to find gcd(a,b) with $a \geq b$,

- n divisions are used to obtain (with $a = r_0$, $b = r_1$):

$$
\begin{aligned}
r_0 &= r_1 q_1 + r_2 && 0 \le r_2 < r_1, \\
r_1 &= r_2 q_2 + r_3 && 0 \le r_3 < r_2, \\
&\;\;\vdots \\
r_{n-2} &= r_{n-1} q_{n-1} + r_n && 0 \le r_n < r_{n-1}, \\
r_{n-1} &= r_n q_n .
\end{aligned}
$$

gcd(108,30)

108 = 30 × 3 + 18
30 = 18 × 1 + 12
18 = 12 × 1 + 6
12 = 6 × 2
GCD

- Since each quotient **$q_1, q_2, \ldots, q_{n-1}$ is at least 1 and $q_n \geq 2$**:

$$
\begin{aligned}
r_n &\ge 1 = f_2, \\
r_{n-1} &\ge 2r_n \ge 2f_2 = f_3, \\
r_{n-2} &\ge r_{n-1} + r_n \ge f_3 + f_2 = f_4, \\
&\;\;\vdots \\
r_2 &\ge r_3 + r_4 \ge f_{n-1} + f_{n-2} = f_n, \\
\boxed{b} = r_1 &\ge r_2 + r_3 \boxed{\ge} f_n + f_{n-1} = \boxed{f_{n+1}}.
\end{aligned}
$$

Lamé's Theorem 2

It follows that if n divisions are used by the Euclidian algorithm to find gcd(a,b) with $a \geq b$, then $b \geq f_{n+1}$. By Example 4, $f_{n+1} > \alpha^{n-1}$, for $n > 2$, where **$\alpha = (1 + \sqrt{5})/2$**. Therefore, **$b > \alpha^{n-1}$**.

Because **$\log_{10} \alpha \approx 0.208 > 1/5$**, $\log_{10} b > (n-1) \log_{10} \alpha > (n-1)/5$. Hence,

$$\boxed{n - 1 < 5 \cdot \log_{10} b.}$$

Suppose that b has k decimal digits. Then $b < 10^k$ and $\log_{10} b < k$. It follows that **$n - 1 < 5k$** and since k is an integer, **$n \leq 5k$**.

As a consequence of Lamé's Theorem, **$O(\log b)$ divisions are used by the Euclidian algorithm to find gcd(a,b) whenever $a > b$.**

- By Lamé's Theorem, the number of divisions needed to find gcd(a,b) with $a > b$ is less than or equal to 5 ($\log_{10} b$ + 1) since the number of decimal digits in b (which equals $\lfloor \log_{10} b \rfloor + 1$) is less than or equal to $\log_{10} b + 1$.

Lamé's Theorem was the first result in computational complexity

Recursively Defined Sets and Structures[1]

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.

Recursively Defined Sets and Structures 2

Example : 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: The natural numbers **N**.

BASIS STEP: $0 \in \mathbf{N}$.

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.

Strings

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

BASIS STEP: $\lambda \in \Sigma^*$ (λ is the empty string)

RECURSIVE STEP: If w is in Σ^* and x is in Σ, then $wx \in \Sigma^*$.

Example: If $\Sigma = \{0,1\}$, the strings in in Σ^* are the set of all bit strings, λ,0,1, 00,01,10, 11, etc.

Example: If $\Sigma = \{a,b\}$, show that aab is in Σ^*.

- Since $\lambda \in \Sigma^*$ and $a \in \Sigma$, $a \in \Sigma^*$.
- Since $a \in \Sigma^*$ and $a \in \Sigma$, $aa \in \Sigma^*$.
- Since $aa \in \Sigma^*$ and $b \in \Sigma$, $aab \in \Sigma^*$.

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 $w \in \Sigma^*$, then $w \cdot \lambda = w$.

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

Often $w_1 \cdot w_2$ is written as $w_1 w_2$.

If $w_1 = abra$ and $w_2 = cadabra$, the concatenation $w_1 w_2 = abracadabra$.

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:

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

Balanced Parentheses

合理的括號

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?

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 $\{\neg, \wedge, \vee, \rightarrow, \leftrightarrow\}$.

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.

Structural Induction

Definition: To prove a property of the elements of a recursively defined set, we use ***structural induction* (結構歸納法)**.

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.

Generalized Induction 1

***Generalized induction* (一般歸納法)** is used to prove results about sets **other than the integers that have the well-ordering property.** (*explored in more detail in Chapter* 9)

For example, consider an ordering on $\mathbf{N} \times \mathbf{N}$, ordered pairs of nonnegative integers. Specify that (x_1, y_1) is less than or equal to (x_2, y_2) if either $x_1 < x_2$, or $x_1 = x_2$ and $y_1 < y_2$. This is called the *lexicographic ordering*.

Strings are also commonly ordered by a *lexicographic ordering*.

The next example uses generalized induction to prove a result about ordered pairs from $\mathbf{N} \times \mathbf{N}$.

Generalized Induction 2

Example: Suppose that $a_{m,n}$ is defined for $(m,n) \in \mathbf{N} \times \mathbf{N}$ by $a_{0,0} = 0$ and

$$a_{m,n} = \begin{cases} a_{m-1,n} + 1 & \text{if } n = 0 \text{ and } m > 0 \\ a_{m,n-1} + n & \text{if } n > 0 \end{cases}.$$

Show that $a_{m,n} = m + n(n+1)/2$ is defined for all $(m,n) \in \mathbf{N} \times \mathbf{N}$.

Solution: Use generalized induction.

BASIS STEP: $a_{0,0} = 0 + (0 \cdot 1)/2$

INDUCTIVE STEP: Assume that $a_{m',n'} = m' + n'(n'+1)/2$ whenever (m',n') is less than (m,n) in the lexicographic ordering of $\mathbf{N} \times \mathbf{N}$.

- If $n = 0$, by the inductive hypothesis we can conclude
  $$a_{m,n} = a_{m-1,n} + 1 = m - 1 + n(n+1)/2 + 1 = m + n(n+1)/2.$$
- If n > 0, by the inductive hypothesis we can conclude
  $$a_{m,n} = a_{m,n-1} + 1 = m + \frac{n(n-1)}{2} + n = m + \frac{n(n+1)}{2}.$$

Chapter 5.
Induction and recursion

Recursive Algorithms

Definition: An algorithm is called ***recursive*** **(遞迴)** 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.

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 $n!$}

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}

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 **mod** a, a)
{output is $gcd(a, b)$}

Recursive Modular Exponentiation Algorithm

Example: Devise a a recursive algorithm for computing b^n **mod** m, where b, n, *and* m are integers with $m \geq 2$, $n \geq 0$, and $1 \leq b \leq m$.

Solution:

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

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, $1 \le i \le j \le n$)
$m := \lfloor (i + j)/2 \rfloor$
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 0}