南方科技大学
SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

# CS215 DISCRETE MATHEMATICS FOR COMPUTER SCIENCE

Dr. Qi Wang

Department of Computer Science and Engineering
Office: Room413, CoE South Tower
Email: wangqi@sustech.edu.cn

# Review

01. Propositional Logic
02. Predicate Logic
03. Mathematical Proofs
04. Sets
05. Functions
06. Complexity of Algorithms
07. Number Theory
    Groups, Rings and Fields
08. Cryptography
09. Mathematical Induction
10. Recursion
11. Counting
12. Relation
13. Graphs
14. Tree

# Review

01. Propositional Logic
02. Predicate Logic
03. Mathematical Proofs
04. Sets
05. Functions
06. Complexity of Algorithms
07. Number Theory
    Groups, Rings and Fields

    Discrete Probability

08. Cryptography
09. Mathematical Induction
10. Recursion
11. Counting
12. Relation
13. Graphs
14. Tree

# Logic

- Logical connectives

# Logic

- Logical connectives

  $\neg p,\ p \vee q,\ p \wedge q,\ p \oplus q,\ p \rightarrow q,\ p \leftrightarrow q$

# Logic

- Logical connectives

  $\neg p,\ p \vee q,\ p \wedge q,\ p \oplus q,\ p \rightarrow q,\ p \leftrightarrow q$

- Logical equivalence

# Logic

- Logical connectives

  $\neg p,\ p \vee q,\ p \wedge q,\ p \oplus q,\ p \rightarrow q,\ p \leftrightarrow q$

- Logical equivalence

  De Morgan's laws, communtative laws, distributive laws, ...

# Logic

- Logical connectives

  $\neg p$, $p \vee q$, $p \wedge q$, $p \oplus q$, $p \rightarrow q$, $p \leftrightarrow q$

- Logical equivalence

  De Morgan's laws, communtative laws, distributive laws, ...

- Predicate logic

  contains variables

# Logic

- Logical connectives

  $\neg p,\ p \vee q,\ p \wedge q,\ p \oplus q,\ p \rightarrow q,\ p \leftrightarrow q$

- Logical equivalence

  De Morgan's laws, communtative laws, distributive laws, ...

- Predicate logic

  contains variables

- Quantified statements

  universal, existential, equivalence

# Methods of Proving Theorems

- Basic methods to prove theorems:
  - *direct proof*
    - $p \to q$ is proved by showing that if $p$ is true then $q$ follows
  - *proof by contrapositive*
    - show the contrapositive $\neg q \to \neg p$
  - *proof by contradiction*
    - show that $(p \land \neg q)$ contradicts the assumptions
  - *proof by cases*
    - give proofs for all possible cases
  - *proof of equivalence*
    - $p \leftrightarrow q$ is replaced with $(p \to q) \land (q \to p)$

- function?

# Set, Function

- function?

  one-to-one (injective) function?

# Set, Function

- function?

  one-to-one (injective) function?

  onto (surjective) function?

# Set, Function

- function?

  one-to-one (injective) function?

  onto (surjective) function?

  bijective function (one-to-one correspondence)?

# Set, Function

- function?

  one-to-one (injective) function?

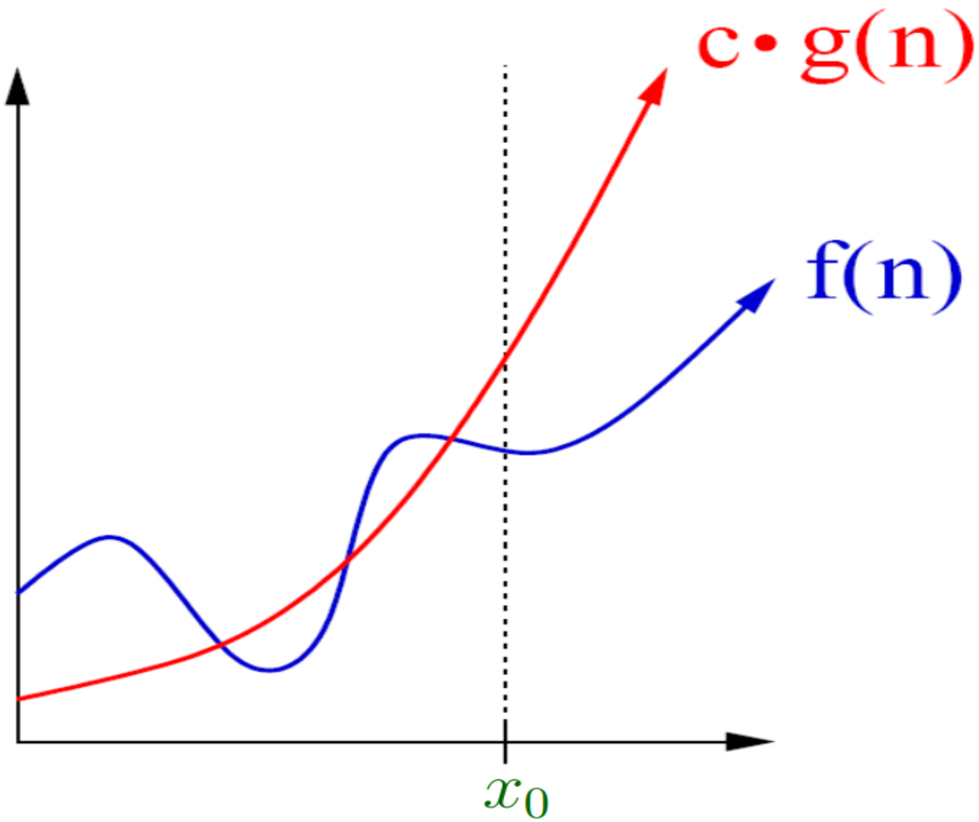
  onto (surjective) function?

  bijective function (one-to-one correspondence)?

- counting the number of such functions?

# Big-$O$ Notation

- Let $f$ and $g$ be functions from the set of integers or the set of real numbers to the set of real numbers. We say that $f(n) = O(g(n))$ (reads: $f(n)$ is $O$ of $g(n)$), if there exist some positive constants $C$ and $k$ such that $|f(n)| \leq C|g(n)|$, whenever $n > k$.



- Divisibility

- Divisibility

  Congruence relation

# Number Theory

- Divisibility

  Congruence relation

  Primes

# Number Theory

- Divisibility

  Congruence relation

  Primes

  GCD and Euclidean Algorithm

- Divisibility

  Congruence relation

  Primes

  GCD and Euclidean Algorithm

  Modular Inverse

- Divisibility

  Congruence relation

  Primes

  GCD and Euclidean Algorithm

  Modular Inverse

    When does an inverse of $a$ modulo $m$ exist?

    How to find inverses?

- Divisibility

  Congruence relation

  Primes

  GCD and Euclidean Algorithm

  Modular Inverse

  When does an inverse of $a$ modulo $m$ exist?

  How to find inverses?

  Chinese Remainder Theorem

# Number Theory

- Divisibility

  Congruence relation

  Primes

  GCD and Euclidean Algorithm

  Modular Inverse

  When does an inverse of $a$ modulo $m$ exist?

  How to find inverses?

  Chinese Remainder Theorem

  Back substitution

- Divisibility

  Congruence relation

  Primes

  GCD and Euclidean Algorithm

  Modular Inverse

  When does an inverse of $a$ modulo $m$ exist?

  How to find inverses?

  Chinese Remainder Theorem

  Back substitution

$$\begin{aligned} x &\equiv 2 \pmod{3} \\ x &\equiv 3 \pmod{5} \\ x &\equiv 2 \pmod{7} \end{aligned}$$

- Fermat's Little Theorem

- Fermat's Little Theorem

  Euler's Theorem

  Primitive roots, multiplicative order

# Cryptography

- Fermat's Little Theorem

  Euler's Theorem

  Primitive roots, multiplicative order

  RSA cryptosystem

  DLP, Diffie-Hellman protocol

# Mathematical Induction

- A *typical* proof by mathematical induction, showing that a statement $P(n)$ is true for all integers $n \geq b$ consists of three steps:

# Mathematical Induction

- A *typical* proof by mathematical induction, showing that a statement $P(n)$ is true for all integers $n \geq b$ consists of three steps:

  1. We show that $P(b)$ is true. – Base Step

- A *typical* proof by mathematical induction, showing that a statement $P(n)$ is true for all integers $n \geq b$ consists of three steps:

1. We show that $P(b)$ is true. – Base Step
2. We then, $\forall n > b$, show either

$$(*) \qquad P(n-1) \to P(n)$$

or

$$(**) \qquad P(b) \wedge P(b+1) \wedge \cdots \wedge P(n-1) \to P(n)$$

- A *typical* proof by mathematical induction, showing that a statement $P(n)$ is true for all integers $n \geq b$ consists of three steps:

1. We show that $P(b)$ is true. – Base Step
2. We then, $\forall n > b$, show either

$$(*) \qquad P(n-1) \rightarrow P(n)$$

or

$$(**) \qquad P(b) \wedge P(b+1) \wedge \cdots \wedge P(n-1) \rightarrow P(n)$$

We need to make the inductive hypothesis of either $P(n-1)$ or $P(b) \wedge P(b+1) \wedge \cdots \wedge P(n-1)$.We then use $(*)$ or $(**)$ to derive $P(n)$.

# Mathematical Induction

- A *typical* proof by mathematical induction, showing that a statement $P(n)$ is true for all integers $n \geq b$ consists of three steps:

1. We show that $P(b)$ is true. – Base Step
2. We then, $\forall n > b$, show either

$$(*) \qquad P(n-1) \rightarrow P(n)$$

or

$$(**) \qquad P(b) \wedge P(b+1) \wedge \cdots \wedge P(n-1) \rightarrow P(n)$$

   We need to make the inductive hypothesis of either $P(n-1)$ or $P(b) \wedge P(b+1) \wedge \cdots \wedge P(n-1)$.We then use $(*)$ or $(**)$ to derive $P(n)$.

3. We conclude on the basis of the principle of mathematical induction that $P(n)$ is true for all $n \geq b$.

# Recurrence

- Iterating a recurrence

# Recurrence

- Iterating a recurrence

  bottom up or top down

# Recurrence

- Iterating a recurrence

  bottom up or top down

  prove by induction, complexity, ...

# Counting

- The sum rule and product rule

# Counting

- The sum rule and product rule

  The Inclusion-Exclusion Principle

# Counting

- The sum rule and product rule

  The Inclusion-Exclusion Principle

  The Pigeonhole Principle

# Counting

- The sum rule and product rule

  The Inclusion-Exclusion Principle

  The Pigeonhole Principle

  **Theorem** If $N$ is a positive integer and $k$ is an integer with $1 \leq k \leq n$, then there are
  $P(n, k) = n(n-1)(n-2)\cdots(n-k+1)$
  $k$-element permutations with $n$ distinct elements.

## Counting

- The sum rule and product rule

  The Inclusion-Exclusion Principle

  The Pigeonhole Principle

  **Theorem** If $N$ is a positive integer and $k$ is an integer with $1 \leq k \leq n$, then there are
  $P(n,k) = n(n-1)(n-2)\cdots(n-k+1)$
  $k$-element permutations with $n$ distinct elements.

  $$P(n,3) = 3! \cdot C(n,3)$$

- The sum rule and product rule

  The Inclusion-Exclusion Principle

  The Pigeonhole Principle

  **Theorem** If $N$ is a positive integer and $k$ is an integer with $1 \leq k \leq n$, then there are
  $P(n,k) = n(n-1)(n-2)\cdots(n-k+1)$
  $k$-element permutations with $n$ distinct elements.

  $P(n,3) = 3! \cdot C(n,3)$

  Pascal's Triangle, Identity

- The sum rule and product rule

  The Inclusion-Exclusion Principle

  The Pigeonhole Principle

  **Theorem** If $N$ is a positive integer and $k$ is an integer with $1 \leq k \leq n$, then there are
  $P(n,k) = n(n-1)(n-2)\cdots(n-k+1)$
  $k$-element permutations with $n$ distinct elements.

  $$P(n,3) = 3! \cdot C(n,3)$$

  Pascal's Triangle, Identity

  The Binomial Theorem, Trinomial

- **Definition** An *$r$-combination* with repetition allowed, or a *multiset of size $r$*, chosen from a set of $n$ elements, is an unordered selection of elements with repetition allowed.

  **Example** Find # multisets of size 17 from the set $\{1, 2, 3\}$.

  This is equivalent to finding the # nonnegative solutions to $x_1 + x_2 + x_3 = 17$.

# Counting

- **Definition** An *r-combination* with repetition allowed, or a *multiset of size $r$*, chosen from a set of $n$ elements, is an unordered selection of elements with repetition allowed.

  **Example** Find # multisets of size 17 from the set $\{1, 2, 3\}$.

  This is equivalent to finding the # nonnegative solutions to $x_1 + x_2 + x_3 = 17$.

- Solving linear (non)homogeneous recurrence relation

# Counting

- **Definition** An *r-combination* with repetition allowed, or a *multiset of size r*, chosen from a set of $n$ elements, is an unordered selection of elements with repetition allowed.

  **Example** Find # multisets of size 17 from the set $\{1, 2, 3\}$.

  This is equivalent to finding the # nonnegative solutions to $x_1 + x_2 + x_3 = 17$.

- Solving linear (non)homogeneous recurrence relation

- Combinatorial proof

# Binary Relations

- Properties of relations

# Binary Relations

- Properties of relations

  Representing relations

# Binary Relations

- Properties of relations

  Representing relations

  Closures on relations

- Properties of relations

  Representing relations

  Closures on relations

  Equivalence relation

  **Definition** A relation $R$ on a set $A$ is called an *equivalence relation* if it is reflexive, symmetric, and transitive.

# Binary Relations

- Properties of relations

  Representing relations

  Closures on relations

  Equivalence relation

  **Definition** A relation $R$ on a set $A$ is called an *equivalence relation* if it is reflexive, symmetric, and transitive.

  Partial ordering

- Properties of relations

  Representing relations

  Closures on relations

  Equivalence relation

  **Definition** A relation $R$ on a set $A$ is called an *equivalence relation* if it is reflexive, symmetric, and transitive.

  Partial ordering

  **Definition** A relation $R$ on a set $A$ is called a *partial ordering* if it is reflexive, antisymmetric, and transitive.

# Graphs & Trees

- Basic concepts

# Graphs & Trees

- Basic concepts

  connected graph, simple graph, isomophism, chromatic number, planar graph, Euler circuit, Hamilton circuit, shortest path, bipartite graph, complete graph, special graphs ($K_n$, $K_{m,n}$, $C_n$, $W_n$, $Q_n$), m-ary tree, tree traversal, spanning tree ...

# Good Luck!