

Unit 3 Discrete Mathematic

Text A About Discrete Mathematic

1. Introduction to Discrete Mathematics

Discrete^① Mathematics is **the general term**^② for several branches of mathematics, which is based on the study of mathematical structures that are fundamentally discrete rather than continuous. **In contrast to**^③ real numbers that have **the property**^④ of varying “smoothly”, the objects studied in discrete mathematics—such as integers, graphs, and statements in logic—do not vary smoothly in this way, but have **distinct**^⑤, separated values.^{注1} Discrete mathematics therefore **excludes**^⑥ topics in “**continuous mathematics**^⑦” such as **calculus**^⑧ and analysis. Discrete objects can often be **enumerated**^⑨ by integers. More formally, discrete mathematics has been characterized as the branch of mathematics dealing with **countable sets**^⑩ (sets that have the same **cardinality**^⑪ as **subsets**^⑫ of the integers, including **rational**^⑬ numbers but not real numbers). However, there is no exact, universally agreed, definition of the term “discrete mathematics.” Indeed, discrete mathematics is described **less** by what is included **than**^⑭ by what is excluded: continuously varying quantities and related **notions**^⑮.

Research in discrete mathematics increased in the latter half of the twentieth century partly due to the development of digital computers which operate in discrete steps and store

-
- ① 离散的
 - ② 一般术语
 - ③ 相反地
 - ④ 性质,性能
 - ⑤ 清楚的,有区别的
 - ⑥ 排除,排斥
 - ⑦ 连续数学
 - ⑧ 微积分学
 - ⑨ 枚举
 - ⑩ 可数集
 - ⑪ 基数
 - ⑫ 子集
 - ⑬ 有理数
 - ⑭ 少于
 - ⑮ 概念

data in discrete **bits**^⑩. Concepts and **notations**^⑪ from discrete mathematics are useful in studying and describing **objects**^⑫ and problems in **branches of**^⑬ computer science, such as computer **algorithms**^⑭, programming languages, **cryptography**^⑮, automated theorem proving, and software development. **Conversely**^⑯, computer **implementations**^⑰ are significant.

2. Topics in discrete mathematics

Theoretical computer science includes areas of discrete mathematics relevant to computing. It draws heavily on graph theory and logic. Included within theoretical computer science is the study of algorithms for computing mathematical results. **Computability**^⑱ studies what can be computed in principle, and has close ties to logic, while **complexity studies**^⑲ the time taken by computations. Automata theory and formal language theory are closely related to computability. Computational geometry applies algorithms to **geometrical**^⑳ problems, while computer image analysis applies them to representations of images. Theoretical computer science also includes the study of various continuous computational topics.

Logic is the study of the principles of valid reasoning and **inference**^㉑, **as well as**^㉒ of **consistency soundness**^㉓, and **completeness**^㉔. For example, in most systems of logic (but not in **intuitionistic logic**^㉕) Peirce's law $((P \rightarrow Q) \rightarrow P) \rightarrow P$ is a **theorem**^㉖. For classical logic, it can be easily **verified**^㉗ with a truth table. The study of mathematical proof is particularly important in logic, and has applications to **automated**^㉘ theorem proving and

-
- ⑩ 位数
 - ⑪ 符号
 - ⑫ 对象,物体
 - ⑬ 分支
 - ⑭ 算法
 - ⑮ 密码学
 - ⑯ 相反地
 - ⑰ 实施
 - ⑱ 可计算性
 - ⑲ 复杂研究
 - ⑳ 几何的
 - ㉑ 推断,推理
 - ㉒ 也
 - ㉓ 一致性,可靠性
 - ㉔ 完整性
 - ㉕ 直觉主义逻辑
 - ㉖ 定理原理
 - ㉗ 验证
 - ㉘ 使自动化

formal **verification**^{③⑤} of software.

Logical formulas are discrete structures, as are proofs, which form finite trees or, more generally, **directed acyclic graphstructures**^{③⑥} (with each inference step combining one or more **premise**^{③⑦} branches to give a single conclusion). The truth values of logical formulas usually form a finite set, generally restricted to two values; true and false, but logic can also be continuous-valued, e. g. , fuzzy logic. Concepts such as infinite proof trees or infinite **derivation**^{③⑧} trees have also been studied, e. g. infinitary logic.

Set theory is the branch of mathematics that studies sets, which are collections of objects, such as {blue, white, red} or the (infinite) set of all prime numbers. **Partially**^{③⑨} ordered sets and sets with other relations have applications in several areas.

In discrete mathematics, countable sets (including finite sets) are the main focus. The beginning of set theory as a branch of mathematics is usually marked by Georg Cantor's work distinguishing between different kinds of infinite set, motivated by the study of trigonometric series^{④⑩}, and further development of the theory of infinite sets is outside the scope of^{④⑪} discrete mathematics.^{注2} Indeed, **contemporary work**^{④⑫} in descriptive set theory makes extensive use of traditional continuous mathematics.

Graph theory, the study of graphs and networks, is often considered part of combinatory, but has grown large enough and **distinct**^{④⑬} enough, with its own kind of problems, to **be regarded as**^{④⑭} a subject in its own right Which **in all areas of**^{④⑮} math and science have extensive application.

Graphs are one of the prime objects of study in Discrete Mathematics. They are among the most **ubiquitous**^{④⑯} models of both natural and human-made structures. They can model many types of relations and process dynamics in physical, biological and social systems. In computer science, they represent networks of communication, data organization, computational devices, the flow of computation, etc. In Mathematics, they are useful in Geometry and certain parts of **topology**^{④⑰}, e. g. **Knot Theory**^{④⑱}. Algebraic graph theory

-
- ③⑤ 验证
 - ③⑥ 有向无环图结构
 - ③⑦ 前提
 - ③⑧ 引出,来历
 - ③⑨ 部分地
 - ④⑩ 三角级数
 - ④⑪ 范围
 - ④⑫ 当代作品
 - ④⑬ 区分
 - ④⑭ 看成为
 - ④⑮ 在所有的领域
 - ④⑯ 普遍存在的
 - ④⑰ 拓扑学
 - ④⑱ 节点理论

has close links with group theory. There are also continuous graphs, however for the most part research in graph theory falls within the **domain**⁴⁹ of discrete mathematics.

Operations research provides techniques for solving practical problems in business and other fields—problems such as allocating resources to maximize profit, or scheduling project activities to minimize risk. Operations research techniques include linear programming and other areas of **optimization**⁵⁰, queuing theory, scheduling theory, network theory. Operations research also includes continuous topics such as **continuous-time Markov process**⁵¹, **continuous-time martingales**⁵², **process optimization**, and **continuous and hybrid**⁵³ control theory.

Although topology is the field of mathematics that **formalize**⁵⁴ and **generalizes**⁵⁵ the **intuitive**⁵⁶ notion of “**continuous deformation**⁵⁷” of objects, it gives rise to many discrete topics; this can be attributed in part to the focus on topological invariants, which themselves usually take discrete values. See **combinatorial topology**⁵⁸, **topological graph theory**⁵⁹, **topological combinatorics**⁶⁰, **computational topology**⁶¹, **discrete topological space**⁶², **finite topological space**⁶³.

Words

algorithm	<i>n.</i> (名词)	算法;算法式(algorithm 的复数)
automate	<i>adj.</i> (形容词)	自动化的;机械化的
bit	<i>n.</i> (名词)	比特,位
calculus	<i>n.</i> (名词)	结石;微积分学
cardinality	<i>n.</i> (名词)	基数;集的势
computability	<i>n.</i> (名词)	可计算性
conversely	<i>adv.</i> (副词)	相反地

-
- 49 领域
 - 50 优化
 - 51 连续时间马尔可夫过程
 - 52 连续时间鞅
 - 53 过程优化及连续混合控制理论
 - 54 形式化
 - 55 一般化
 - 56 直觉
 - 57 连续变形
 - 58 组合拓扑
 - 59 拓扑图论
 - 60 拓扑组合
 - 61 计算拓扑
 - 62 离散空间
 - 63 有限拓扑空间

cryptography	<i>n.</i> (名词)	密码学;密码使用法
derivation	<i>n.</i> (名词)	引出;来历;词源
discrete	<i>adj.</i> (形容词)	离散的,不连续的
distinct	<i>adj.</i> (形容词)	明显的;独特的;清楚的;有区别的
domain	<i>n.</i> (名词)	领域;域名;产业;地产
enumerate	<i>n.</i> (名词)	枚举类型
exclude	<i>vt. & vi.</i> (动词)	排除;排斥;拒绝接纳;逐出
formalize	<i>vt. & vi.</i> (动词)	使形式化;使正式;拘泥礼仪
generalize	<i>vt. & vi.</i> (动词)	概括;推广;使……一般化
geometrical	<i>adj.</i> (形容词)	几何的,几何学的
hybrid	<i>n.</i> (名词)	混合
implementation	<i>n.</i> (名词)	实施
inference	<i>n.</i> (名词)	推理;推论;推断
intuitive	<i>adj.</i> (形容词)	直觉的;凭直觉获知的
notation	<i>n.</i> (名词)	符号;记法
notion	<i>n.</i> (名词)	观念;小商品
object	<i>n.</i> (名词)	物体;对象
optimization	<i>n.</i> (名词)	最佳化,最优化
partially	<i>adv.</i> (副词)	部分地;偏袒地
premise	<i>vt. & vi.</i> (动词)	引出,预先提出;作为……的前提
property	<i>n.</i> (名词)	性质,性能;财产;所有权
rational	<i>adj.</i> (形容词)	合理的;理性的
subsets	<i>n.</i> (名词)	子集合(subset 的复数)
theorem	<i>n.</i> (名词)	定理原理
topology	<i>n.</i> (名词)	拓扑学;地质学;局部解剖学
ubiquitous	<i>adj.</i> (形容词)	普遍存在的;无所不在的
verification	<i>n.</i> (名词)	确认,查证;核实
verified	<i>adv.</i> (副词)	已查清的,已证实的

Phrases

as well as	也;和……一样;不但……而且
be regarded as	被认为是;被当作是
combinatorial topology	组合拓扑学;组合拓扑
complexity studies	复杂性研究
computational topology	计算拓扑
consistency soundness	一致性,可靠性
contemporary work	当代作品

continuous deformation	连续形变
continuous mathematics	连续数学
continuous-time Markov process	连续时间的马尔可夫过程
continuous-time martingales	连续时间鞅
countable sets	可数集
directed acyclic graphstructures	有向无环图结构
discrete topological space	离散拓扑空间
distinct enough	截然不同
finite topological space	有限拓扑空间
in all areas of	在各方面的
in branches of	分支的
in contrast to	相比之下
intuitionistic logic	直觉主义逻辑
knot theory	节点理论;[数] 纽结理论
the general term for	一般的术语
the scope of	范围
topological combinatorics	拓扑组合
topological graph theory	图论拓扑
trigonometric series	三角级数

Exercises

【Ex1】 Answer the questions according to the text:

- (1) What is Discrete Mathematic?
- (2) Why did Discrete Mathematic develop so fast in the twentieth century?
- (3) How many topics are there in this chapter, and what are they?
- (4) What are the logic formulas?
- (5) What does Operations research involve?

【Ex2】 Translate into Chinese:

- (1) Graph theory is an old subject with modern applications.
- (2) Relation between elements of sets is represented using the structure called a relation.
- (3) Much of discrete mathematic is developed of discrete structures, which are used to represent discrete objects.
- (4) Discrete mathematic is the gateway to more advanced courses in all parts of the mathematical sciences.

- (5) The computer chip is primarily responsible for executing instructions.
- (6) Tape must be read or written sequentially, not randomly.
- (7) Deselect the text by clicking anywhere outside of the selection on the page or pressing an arrow key on the keyboard.
- (8) Faster than many types of parallel port, a single USB port is capable of chaining many devices without the need of a terminator.

【Ex3】 Choose the best answer:

- (1) Very long, complex expressions in program are difficult to write correctly and difficult to _____.
 A. defend B. detect C. default D. debug
- (2) _____ is the study of the principles of valid reasoning and inference, as well as of consistency, soundness, and completeness.
 A. Graph theory B. Logic
 C. Topology D. Operation research
- (3) The _____ storage area that you can use to copy or move selected text or object among applications.
 A. exponent B. order C. temporary D. superior
- (4) Software design is a _____ process. It requires a certain _____ of flair on the part of the designer.
 A. create B. created C. creating D. creative
 A. amount B. amounted C. mount D. mounted

批 注

注1 In contrast to 短语引导的独立结构;that have the property of varying “smoothly”的先行词为 real numbers,属定语从句;the objects 为句子的主语;studied 为谓语;do not vary smoothly in this way, but have distinct, separated values. 补充说明。

注2 The beginning of set theory 为主语;as a branch of mathematics 为方式状语;is marked, motivated,and further development of the theory of infinite sets 为句子谓语。

Text B Tree

A **connected graph**^① that contains no simple **circuits**^② is called a tree. Trees were used as long ago as 1857, when the English mathematician Arthur Cayley used them to count

① 连通图

② 电路,回路

certain types of chemical **compounds** ③. Since that time, trees have been employed to solve problems **in a wide variety of** ④ **disciplines** ⑤.

Tree are particularly useful in computer science, for instance, trees **are employed to** ⑥ construct efficient algorithms for locating **items** ⑦ in a list. They are used to construct network with the least expensive set of telephone lines storing and **transmitting** ⑧ data. ① Trees can model procedures that **are carried out** ⑨ using a sequence of decisions. This makes trees valuable **in the study** ⑩ of sorting algorithms.

1. Definitions

A tree is an undirected simple graph G that satisfies any of the following **equivalent** ⑪ conditions ②:

- G is connected and has no cycles.
- G has no cycles, and a simple cycle is formed if any edge is added to G .
- G is connected, and it is not connected anymore if any edge is removed from G .
- G is connected and the 3-vertex complete graph K_3 is not a minor of G . ③
- Any two **vertices** ⑫ in G can be connected by a unique simple path.

If G has finitely many vertices, say n of them, then the above statements are also equivalent to any of the following conditions:

- G is connected and has $(n-1)$ edge.
- G has no simple cycles and has $(n-1)$ edge.

Which of the graphs shown in Figure 3-1 are trees?

G_1 and G_2 are trees, since both are connected graphs with no simple circuits; G_3 is not a tree because $e . b . a . d . e$ is a simple circuit. ④

In this graph. Finally, G_1 is not a tree since it is not connected.

2. Facts

- (1) A tree with n vertices has $(n-1)$ edges.
- (2) A full m -ary tree with i internal vertices contains $n = m * i + 1$ vertices.
- (3) A full m -ary tree with.

③ 混合物
④ 以更广泛的
⑤ 学科,分支
⑥ 用于
⑦ 条款,项目
⑧ 传播;发射
⑨ 执行
⑩ 研究
⑪ 等价的
⑫ 节点

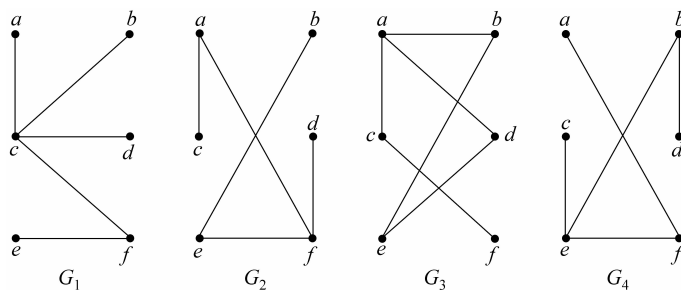


Figure 3-1 Graphs

(i) n vertices has $i = (n - 1)/m$ **internal**^⑮ vertices and $L = [(m - 1) * n + 1]/m$ leaves.

(ii) i internal vertices has $n = m * i + 1$ vertices and $L = (m - 1) * i + 1$ leaves.

(iii) L leaves has $n = (m * L - 1)/(m - 1)$ vertices and $i = (L - 1)/(m - 1)$ internal vertices.

3. Tree spices

(1) M-ary tree is a tree with the property that every internal **vertex**^⑯ has no more than m children.

(2) Binary tree is an m-ary tree with $m = 2$ (each child may be designated as a left or a right child of its parent).

(3) Ordered tree is a tree in which the children of each internal vertex are **linearly**^⑰ ordered.^⑸

(4) Balance tree is a tree in which every vertex is at level h or $h-1$, where h is the height of the tree.^⑹

(5) Binary search tree is a binary tree in which the vertices are labeled with items **so that**^⑱ a label of a vertex is greater than the labels of all vertices in the left subtree of this vertex is an less than the label of all vertices in the right sub tree of this vertex.

(6) Decision tree is a rooted tree where each vertex represents a possible outcome of a decision and the leaves represent the possible solutions.

(7) **Spanning tree**^⑲: a tree containing all vertices of a graph.

(8) **Minimum spanning tree**^⑳: a spanning tree with smallest possible sum of weights of its edges.^⑺

⑮ 内部的

⑯ 节点

⑰ 呈直线地

⑱ 致使

⑲ 生成树

⑳ 最小生成树

4. Tree traversal algorithms

Procedures for systematically visiting every vertex of an ordered rooted tree are called **traversal algorithms**^①. We will describe three of the most commonly used such algorithms: preorder traversal, inorder traversal, and postorder traversal.

(1) Preorder traversal

Preorder traversal^② is listing of the vertices of an ordered rooted tree defined recursively by specifying that the root is listed. Followed by the first subtree, followed by the other sub trees in the order they occur from left to right.

Let T be an ordered rooted tree with root r . If T consists only of r , then r is the preorder traversal of T . Otherwise, suppose that $T_1, T_2, \dots, T_n$ are the **sub trees**^③ at r from left to right in T . The preorder traversal begins by visiting r . It continues by traversing T_1 in preorder, then T_2 in preorder, and so on, until T_n is traversed in preorder.

(2) Inorder traversal

Inorder traversal^④ is a listing of^⑤ the vertices of an ordered rooted tree defined recursively by^⑥ specifying that the first sub tree is listed **followed by**^⑦ the root, followed by the other sub trees in the order they occur from left to right.

Let T be an ordered rooted tree with root r . If T consists only of r , then r is the inorder traversal of T . Otherwise, suppose that $T_1, T_2, \dots, T_n$ are the subtrees at r from left to right in T . The inorder traversal begins by traversing T_1 in inorder. Then visiting r , It continues by traversing T_2 in inorder, then T_3 in inorder, ..., And finally T_n in inorder.

(3) Postorder traversal

Postorder traversal^⑧ is a listing of the vertices of an ordered rooted tree defined recursively by specifying that the sub trees are listed in the order they occur from left to right, followed by the root.^{注8}

Let T be an ordered rooted tree with root r if T consists only of r , then r is the inorder traversal of T . Otherwise, suppose that $T_1, T_2, \dots, T_n$ are the subtrees at r from left to right. The postorder traversal begins by traversing T_1 in postorder, then T_2 in postorder, ..., Then T_n in postorder, and ends by visiting r .

① 遍历算法

② 前序遍历

③ 子树

④ 中序遍历

⑤ 列表

⑥ 被定义为

⑦ 紧跟着

⑧ 后序遍历