

# Discrete Mathematics

## (이산 수학)

국민대학교 컴퓨터 공학부

우종우

[cwwoo@kookmin.ac.kr](mailto:cwwoo@kookmin.ac.kr)

# Announcement

- 과목 홈 페이지 : 종합정보시스템 가상대학(eCampus)
- 교재: Discrete Mathematics, Johnsonbaugh, Pearson출판사, 2009
  - 기타: 이산수학 참고서, 강의노트(가상대학)
- 평가: 중간(40%), 기말(40%), 과제(15%), 출석(5%)
- 수업목차:
  - Introduction, Logic and Proof
  - Set theory, Relations and Function
  - Algorithm, counting method,
  - Graph and Tree
  - Automata

# What is Discrete Mathematics?

- Mathematics dealing with discrete structures
- **discrete(이산) mathematics:**
  - discrete = composed of distinct, separable parts
  - discrete process (분리 또는 불연속) ex) bank transactions
  - countable, finite (예: 디지털 시계, integers, statement in logic)
  - ➔ study of mathematical structures and objects that are fundamentally **discrete** rather than **continuous**
- **Continuous(연속) mathematics:**
  - continuous process (ex. 대수학- differential equations, 아날로그 시계, real numbers)
  - uncountable, continuous representations

# Discrete Math is essential to Computer Science

- Various aspects of discrete mathematics form the foundations for:
  - 1) Modeling Computer Structures (컴퓨터 데이터는 이산적)
  - 2) Designing programs and algorithms (컴퓨터시스템과 프로그램 설계에 있어서 발생하는 제반 문제를 해결하는 기반을 제공)
  - 3) **Reasoning** about programs and Algorithms  
(ex. Logical, relational, recursive, quantitative analytical thinking)

ex)

- 시스템 구축 - 수학적 모델링 으로 이루어짐
- 문제 해결 - 수학의 논리적 개념 이용
- 자료 저장 - 그래프, 트리 등 개념 이용
- 언어 관련 - 집합의 개념 사용

# Applications of discrete mathematics

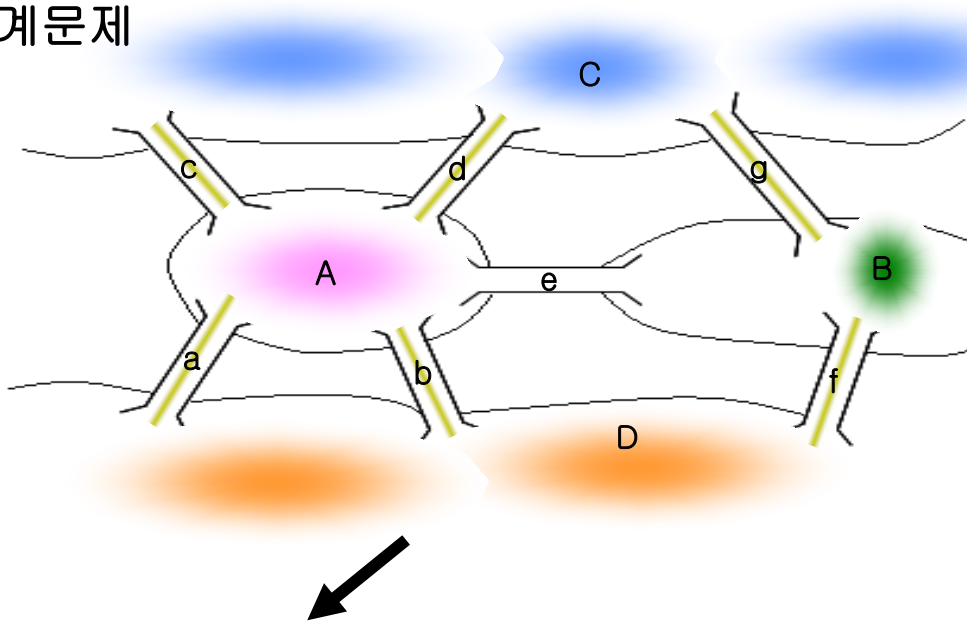
- How can a circuit that adds two integers be designed?
- How many ways are there to choose a valid password on a computer?
- What is the shortest path between two cities using transportation system?
- How can I encrypt a message so that no unintended recipient can read it?
- How many valid internet addresses are there?
- How can a list of integers be sorted so that the integers are in increasing order?

# Uses for Discrete Math in Computer Science

- Advanced algorithms & data structures
- Programming language compilers & interpreters.
- Computer networks
- Operating systems
- Computer architecture
- Database management systems
- Cryptography
- Graphics & animation algorithms, game engines, *etc....*

# Example – Graph Algorithm

실세계문제



Possible Applications:

- routing problem, max flow problem, road networks, equipment replacement,...

최단거리 구하기 알고리즘

**Procedure dijkstra (w, a, z, L)**

**L(a) := 0**

**for all vertices  $x \in a$  do**

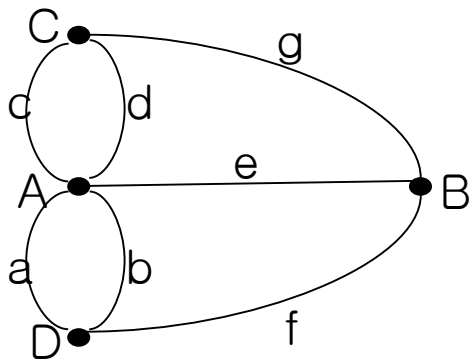
**L(x) =  $\infty$**

**T := set of all vertices**

**...**

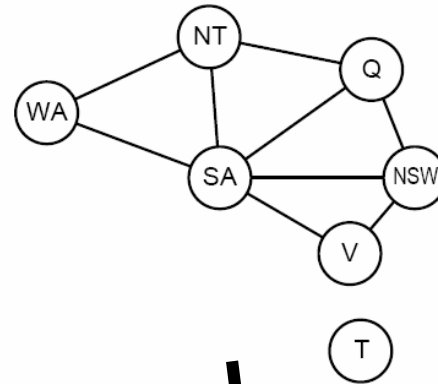
**While z do**

**.....**



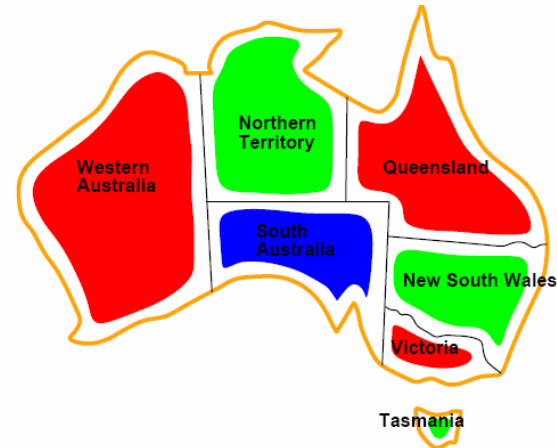
그래프모델

# Graph coloring example



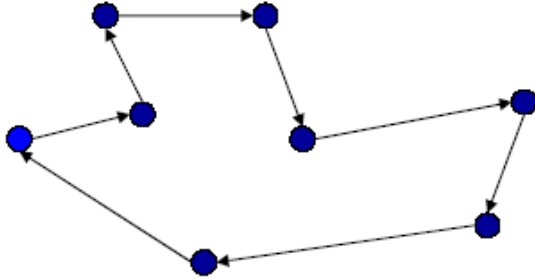
**How to color this map so that no two adjacent regions have the same color?**

=> What's the minimum number of colors such that any two nodes connected by an edge have different colors?





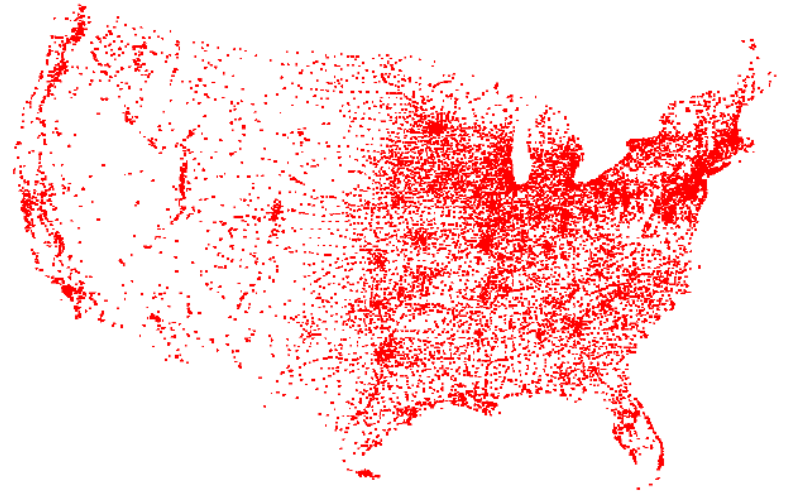
# Traveling Salesman Problem (TSP)



Find a closed tour of minimum length visiting all the cities.

TSP: lots of applications

- Transportation related: scheduling deliveries
- Many others: ex) Scheduling of a machine to drill holes in a circuit board , Genome sequencing, etc



13,509 cities in the US

$13508! = 1.4759774188460148199751342753208e+49936$

# Probability: Bayesian Reasoning

- Bayesian networks provide a means of expressing joint probability over many interrelated hypotheses and therefore reason about them.
- **Example of Query: what is the most likely diagnosis for the infection given all the symptoms?**
- Bayesian networks have been successfully applied in diverse fields such as medical diagnosis, image recognition, language understanding, search algorithms, and many others.

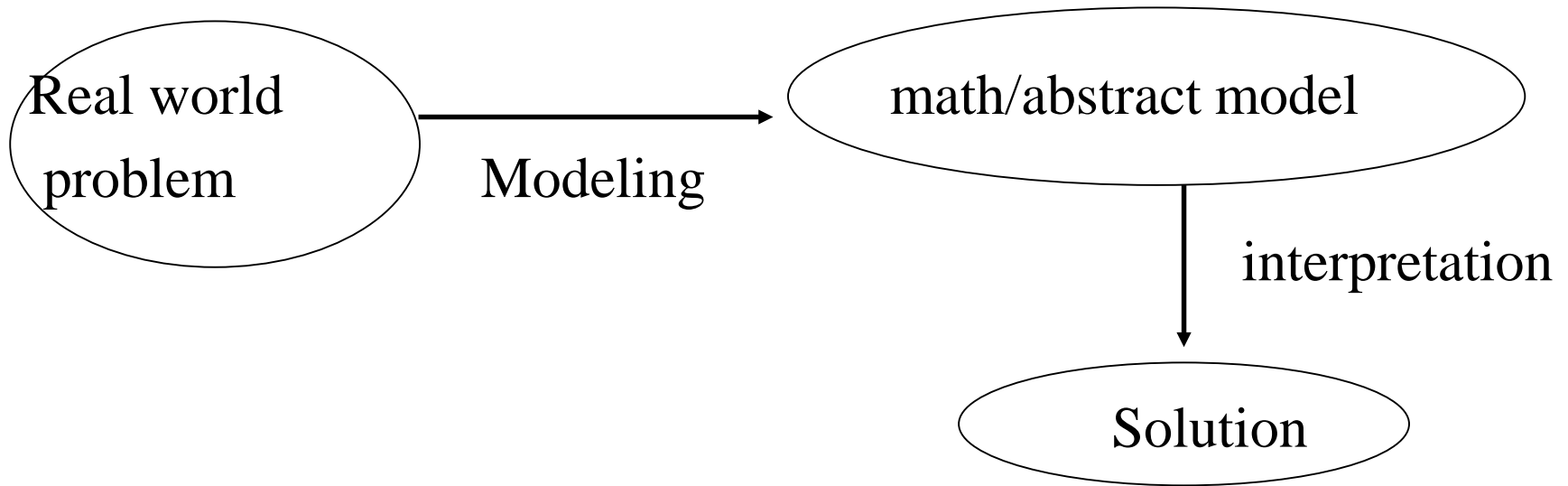
$$P(\mathbf{x}|\omega_i) = \frac{P(\omega_i|\mathbf{x})P(\mathbf{x})}{P(\omega_i)}$$

Bayes Rule

# Modeling for Problem-Solving

\*Mathematical modeling - General methodology using mathematics for problem solving

- 비유된 형태로 쉽게 정보를 이해, 조사와 예측이 용이
- 필요요건: 정의, 문장, 명제, 공리, 연산자, 추론,...등



# Modeling for Problem-Solving

Ex) Jack has twelve apples, which cost him 10 cents apiece.

How many apples must he sell at 20 cents apiece before he begins to make a profit.

- Math model:

expense = (cost) \* (number bought)

(sale price) \* (x) = expense

$$20x = 10 * 12$$

- solution:  $x=6$

# Modeling for Problem-Solving

- **Math model:**

expense = (cost) \* (number bought)

(sale price) \* (x) = expense

- **Algorithm**

begin

input cost, number

expense = cost \* number     { cost=10cents, number = 12 }  
{ cost=10cents, number = 12 }

x = expense / sale-price     { \* sale price = 20 cents \* }

output x

end

- **Coding**

# Modeling for Problem-Solving

## **\*Algorithm**

- a precise and unambiguous sequence of instruction that leads to the solution of a problem in a finite amount of time
- we can approach the problem by attempting to find solution in forms of an algorithm
- Common techniques in CS
  - 1) Obtain math. model
  - 2) formulate algorithm in terms of modeling
  - 3) and then generate S/W

# Topics of Discrete Mathematics

- Logic and Proofs
  - propositions, Predicates, Proofs
- Sets
- Relations & Functions
- Algorithms
- Counting methods
  - Permutations, Combinations
- Graphs
- Trees
- Automata
- Sorting
- Boolean Algebra
- etc