Course Information (I)

- Class homepage
  - http://iphotonics.dyndns.org/dwhuang/courses/cs/
- Time/Location: Tuesday 5, Wednesday 7, MD 231
- Instructor: Ding-wei Huang (dwhuang@cc.ee.ntu.edu.tw) Office: MD 720
- TA:
  - Office hour:
- Credits: 3 units

Course Information (II)

- Grading
  - Midterm exam: 35%
  - Final exam: 40%
  - Homework: 25%
  - Project Bonus: 1-5%

- Some other notes
  - Homework and its due date will be announced on the class homepage.
  - Please hand-in your homework on time. For the delay of your homework, 20 points/day will be subtracted till zero.
Chapter 0: Introduction

- 0.1 The Role of Algorithms
- 0.2 The Origins of Computing Machines
- 0.3 The Science of Algorithms
- 0.4 Abstraction
- 0.5 An Outline of Our Study
- 0.6 Social Repercussions

Terminology

- **Algorithm**: A set of steps that defines how a task is performed
- **Program**: A representation of an algorithm
- **Programming**: The process of developing a program
- **Software**: Programs and algorithms.
- **Hardware**: Equipment

Figure 0.1 An algorithm for a magic trick

Effect: The performer places some cards from a normal deck of playing cards face down on a table and mixes them thoroughly while spreading them out on the table. Then, as the audience requests either red or black cards, the performer turns over cards of the requested color.

Secret and Patter:

**Step 1.** From a normal deck of cards, select ten red cards and ten black cards. Deal these cards face up on the table in any order.

**Step 2.** Announce that you have selected some red cards and some black cards.

**Step 3.** Pick up the red cards. Under the pretense of aligning them into a neat stack, flip them face down in your right hand, palm them, and turn them over again in your right hand, palm them, and turn them over again, etc. Then place the deck of red cards face down on the table as you say, “Here are the red cards in this stack.”

**Step 4.** Pick up the black cards. In a manner similar to that in Step 3, give these cards a slight hesitation as you place them face down in your right hand, palm them, etc. Then place the deck of black cards face down on the table as you say, “Here are the black cards in this stack.”

**Step 5.** In a manner similar to that in Step 3, give the stack of red cards a slight hesitation. Then place the stack on the table as you say, “Here are the red cards in this stack.”

**Step 6.** As long as there are face-down cards on the table, repeatedly execute the following steps:

6.1. Ask the audience to request either a red or a black card.

6.2. If the color requested is red and there is a two-down card with a red on it, use your right hand to turn over the card and say, “Here is a red card.”

6.3. If the color requested is black and there is a two-down card with a black on it, use your right hand to turn over the card and say, “Here is a black card.”

6.4. Otherwise, state that there are no more cards of the requested color and turn over the remaining cards to prove your claim.
History of Algorithms

• The study of algorithms was originally a subject in mathematics.
• Early examples of algorithms
  – Long division algorithm
  – Euclidean Algorithm
• Gödel's Incompleteness Theorem: Some problems cannot be solved by algorithms.

Origins of Computing Machines

• Early computing devices
  – Abacus: positions of beads represent numbers
  – Gear-based machines (1600s-1800s)
    • Positions of gears represent numbers
    • Blaise Pascal, Wilhelm Leibniz, Charles Babbage

Figure 0.2 The Euclidean algorithm

Description: This algorithm assumes that its input consists of two positive integers and proceeds to compute the greatest common divisor of these two values.

Procedure:
Step 1. Assign M and N the value of the larger and smaller of the two input values, respectively.
Step 2. Divide M by N, and call the remainder R.
Step 3. If R is not 0, then assign M the value of N, assign N the value of R, and return to step 2; otherwise, the greatest common divisor is the value currently assigned to N.

Figure 0.3 An Abacus
Early Data Storage

- Punched cards
  - First used in Jacquard Loom (1801) to store patterns for weaving cloth
  - Stored programs in Babbage’s Analytical Engine
  - Popular through the 1970’s
- Gear positions

Early Computers

- Based on mechanical relays
  - 1940: Stibitz at Bell Laboratories
  - 1944: Mark I: Howard Aiken and IBM at Harvard
- Based on vacuum tubes
  - 1937-1941: Atanasoff-Berry at Iowa State
  - 1940s: Colossus: secret German code-breaker
  - 1940s: ENIAC: Mauchly & Eckert at U. of Penn.

Figure 0.4 The Mark I computer

Personal Computers

- First used by hobbyists
- IBM introduced the PC in 1981
  - Accepted by business
  - Became the standard hardware design for most desktop computers
  - Most PCs use software from Microsoft
Computer Science

• The science of algorithms
• Draws from other subjects, including
  – Mathematics
  – Engineering
  – Psychology
  – Business Administration
  – Psychology

Central Questions of Computer Science

• Which problems can be solved by algorithmic processes?
• How can algorithm discovery be made easier?
• How can techniques of representing and communicating algorithms be improved?
• How can our knowledge of algorithms and technology be applied to provide better machines?
• How can characteristics of different algorithms be analyzed and compared?

Figure 0.5 The central role of algorithms in computer science

Abstraction

• **Abstraction**: The distinction between the external properties of an entity and the details of the entity’s internal composition
• **Abstract tool**: A “component” that can be used without concern for the component’s internal properties
Outline of Our Study

• Chapter 1: Data Storage
• Chapter 2: Data Manipulation
• Chapter 3: Operating Systems
• Chapter 4: Networks and the Internet
• Chapter 5: Algorithms
• Chapter 6: Programming Languages

Outline of Our Study (continued)

• Chapter 7: Software Engineering
• Chapter 8: Data Abstractions
• Chapter 9: Database Systems
• Chapter 10: Computer Graphics
• Chapter 11: Artificial Intelligence
• Chapter 12: Theory of Computation

Social Repercussions

• Advances in computer science raise new questions.
  – In law: Questions of rights and liabilities
  – In government: Questions of regulation
  – In the work place: Questions of professionalism
  – In society: Questions of social behavior

Ethical Theories

• Consequence based:
  What leads to the greatest benefit?
• Duty based:
  What are my intrinsic obligations?
• Contract based:
  What contracts must I honor?
• Character based:
  Who do I want to be?