國立臺灣大學
National Taiwan University

EE4033 901/39000.02 ALGORITHMS

Iris Hui-Ru Jiang

Fall 2017

Department of Electrical Engineering
National Taiwan University

Administrative Matters

- **Time/location:** Thursdays 14:20~17:10/EEII-146
- **Instructor:** 江蕙如 Iris Hui-Ru Jiang
  - Email: huiru.jiang@gmail.com
  - Office: BL-629. Tel: 3366-4690
  - Office hours: Thursdays 13:30~14:00, made by appointment
- **Teaching assistant:** Yu-Sheng Lu
  - Email: yslu@eda.ee.ntu.edu.tw
  - Lab: BL-406. Tel: 3366-3700 #6406
  - Office hours: Mondays 13:30~14:30
- **Prerequisites:** two out of the following courses
  - Data structures
  - Discrete mathematics
  - Computer programming in C
  - Computer programming in C++
  - **C/C++ programming skill is a must**

Reading Materials

- Course webpage:
  - https://ceiba.ntu.edu.tw/1061EE4033_02
- Required text:
  - Kleinberg and Tardos, *Algorithm Design*, Addison Wesley, 2006
    - Jon Kleinberg, 20 Best Brains under 40, Discover Magazine, 2008
    - Cornell
- References:
  - Dasgupta, Papadimitriou, and Vazirani, *Algorithms*, McGraw-Hill, 2007
    - UC Berkeley
  - Cormen, Leiserson, Rivest, Stein, *Introduction to Algorithms*, 3rd Ed., McGraw Hill/MIT Press, 2009
    - Bible! MIT

Course Objectives

1. Study unifying principles and concepts of algorithm design
   - Algorithmic problems form the heart of computer science
2. Polish your critical thinking and problem-solving technique
   - Algorithmic problems tend to come bundled together with lots of messy, application-specific detail, some of it essential, some of it extraneous
   - Two fundamental components
     - Get to the mathematically clean core of a problem
     - Identify the appropriate algorithm design techniques based on the structure of the problem
3. Have fun!

- Intended audience:
  - Who are interested in computer science
  - Who are computing something
  - Who are learning problem-solving techniques

Course Content (1/3)

- Introduction
  - An opening problem: stable marriages
  - Range of problems we will consider
- Background:
  - Basics of algorithm analysis
  - Graphs
- General algorithmic techniques
  - Greedy algorithms: Finding optimal solutions with greedy methods
    - Scheduling time intervals
    - The minimum spanning tree problem
  - The divide and conquer method
    - Some basic primitives in computational geometry

*The details are subject to change

Course Content (2/3)

- Dynamic programming with many applications
  - Weighted interval scheduling
  - Knapsack problems
  - Shortest paths
  - Sequence alignment
    - Including efficient implementation via divide and conquer
- Flows and cuts in networks
  - The basic flow and cut problems
  - Basic methods: augmenting paths
  - Application to matching
  - Polynomial time methods
  - Extensions to more general models
  - Applications to resource allocation, sequencing, and segmentation

*The details are subject to change

Course Content (3/3)

- Computational intractability
  - NP-completeness
    - Hardness of problems in optimization and constraint satisfaction
    - How to show NP-completeness: reducibility
  - PSPACE completeness (optional)
    - Hardness of problems in artificial intelligence and game-playing
- Advanced techniques
  - Amortized analysis
  - Linear programming
  - Other topics as time permits: matching algorithms, approximation algorithms, randomized algorithms

*The details are subject to change

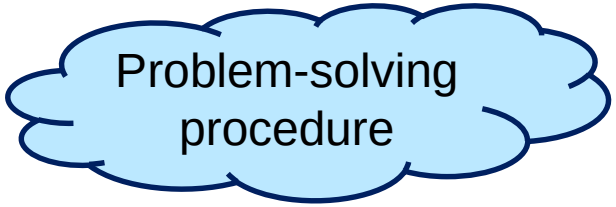
Grading Policy (1/2)

- Grading:
  - Homework: 10%
  - **Programming** projects: (2 mini-projects: 20% + term project: 20%)
  - Tests: (Midterm on Nov. 16: 25% + Final on Jan. 4: 25%)
  - Adjustment: +-5% for each item
- Attention:
  - The grades on homework, projects, and tests are considered **final one week** after they have been handed back, so you should bring any questions to the grader's attention promptly.
  - The final grade is **not negotiable** except instructor's mistakes.
- Academic Honesty: Plagiarism is strongly prohibited.
  - Oral discussion about homework is not considered cheating. Copying someone else's homework/test or part of it is cheating. When cheating is discovered, all students involved will receive no credit for the homework/test, possibly an F grade for the course.

Grading Policy (2/2)

- Homework:
  - Students may discuss the homework problems with one another but must write up their solutions separately.
  - Homework must be handed in at the **beginning** of the class on which it is due.
  - **Late homework will not be accepted**.
- Project:
  - All submissions of mini-projects and term project will be subject to duplication checking; those with ≥ 40% similarity will be penalized.
  - Late submission will incur a penalty of 1/86400 of the total score per second after the deadline (the penalty will be computed based on the submission time).
  - Term project: form 2-person teams, give presentations and submit programs on Jan. 11; a 1-page project proposal is due on Nov. 9.

What is an Algorithm?



- Definition: An algorithm is
  - A finite, definite, effective procedure, with some output [Donald Knuth, 1968]
  - Input: may have
  - Output: must have
  - Definiteness: must be clear and unambiguous
  - Finiteness: terminate after a finite number of steps
  - Effectiveness: must be basic and feasible with pencil and paper
  - Procedure: the sequence of specific steps in a logical order

- Cf. An algorithm is
  - A well-defined procedure for transforming some input to a desired output [Cormen et al. Introduction to Algorithms, 2nd Ed.]