

# CSIE1212: Data Structures and Algorithms

Michael Hsin-Mu Tsai and Hsuan-Tien Lin

Dept. of CSIE, NTU

Course Introduction, February 23, 2021

# Three Warnings Before (Signing for) the Course (1/3)

## 警告: High Expectations

- goal of NTU DSA class:  
as good as the best ones in the world
- tentatively, 3 homework sets, midterm exam, final exam, AND final project (<https://cool.ntu.edu.tw/courses/4621>)
- writing assignments and **time-consuming** programming assignments
- even with **homework 0** now—self-graded but **important**

be prepared to **work hard!**

## Three Warnings Before (Signing for) the Course (2/3)

### 警告: Strict Instructors

- Will you give me a second chance if I copy homework from other people? No.
- Could you let me pass because I will be kicked out by the 1/2 rule? No.
- Will you change my score from F to C? No.

be prepared to **follow the rules!**

## Three Warnings Before (Signing for) the Course (3/3)

### 警告: Uncertain Outcome

- first time **co-teaching** a super big class together
- needing to **experiment** to **scale**
  - multi-room screencast teaching**, for instance
- How many people will not pass?  
**We don't know yet.**
- Will your investment (time) get good return (knowledge)?  
**No guarantees, but we'll try our best.**

be prepared to **take some risks!**

## Wise Words

給資訊系的同學們：努力加油  
給想加選的同學們：審慎考慮

# Some Historical Notes

*A while ago, when I was (we were) a freshman in NTU CSIE (1997).....*

- 「計程」有兩學期，上學期教C，下學期教C++
- 大二上學期教「資料結構」
- 大二下學期教「演算法」

*Then, in my senior year (2001).....*

- 「計程」變成一學期，大一下學期教「物件導向程式設計」(Java)
- 大二上學期教「資料結構與演算法上」
- 大二下學期教「資料結構與演算法下」

*Then, starting 2010.....*

- 物件導向程式設計變為選修
- 大一下學期教「資料結構與演算法」
- 大二上學期教「演算法設計與分析」

# Reasons

- 兩學期的「計程」變成一學期、「物件導向程式設計」變成選修：  
相信同學們可以有自己學習不同語言的能力。
- 把「資料結構」及「演算法」合成一門課：  
兩者互相依賴，其實不容易分散來教。
- 把「資料結構與演算法上/下」區分成「資料結構與演算法」和  
「演算法設計與分析」：
  - 前者以實作為主，銜接計程做更深入的程式練習
  - 後者以分析為主，建立在前者的基礎上探討更多不同的演算法

# Co-Teaching Information

- instructors:  
蔡欣穆 Michael Hsin-Mu Tsai (hsinmu@csie.ntu.edu.tw)  
林軒田 Hsuan-Tien Lin (htlin@csie.ntu.edu.tw)
- course webpage, announcement, discussion forum:  
[https://cool.ntu.edu.tw/courses/4621/ \(NTU COOL\)](https://cool.ntu.edu.tw/courses/4621/)

**rule of thumb to facilitate passing:  
interact with us frequently through NTU COOL!**

# Co-Teaching: Enrollment and Classrooms

- type 1: effectively unlimited
- main classroom: CSIE R103
- other classrooms
  - CSIE R102, R104
  - anywhere with an internet, like your dorm room!
- auditing:
  - welcomed to access NTU COOL, fill here to apply:  
<https://forms.gle/2RSEB2P59oRpYveZ9>



- welcomed to sit only if there is an empty chair

**please think about the **heavy** loading  
before you choose to enroll**

## In-class Interaction - Slido

Connect to <https://sli.do>, Event code: #F061

- Interact with the instructor! **Comment** or **leave questions** about materials just taught.
- Contribute by **upvote** or **downvote** - those with *higher vote counts* will be responded first
- Comments will be monitored and moderated before moving to public area
- Instructor would frequently check slido during the class and respond in real time
- Use your real name — be responsible to your own comments / questions

advice: start **learning to ask questions** as soon as you can

# Plans for the First Half

course time: Tuesdays 13:20–16:10

## Instructor: Hsuan-Tien

- Youtube screencast + **slido interaction**
- 15-min break liberally in the middle (more if needed)
- office hour after class or by appointment

## Topics

- Course Introduction
- Arrays and Linked Lists
- Performance Evaluation
- Stacks and Queues
- Heap
- Tree
- Sorting

## about Hsuan-Tien: Strict but Friendly

- Will you repeat the previous code/slide again? Yes.
- Will you discuss with me after class if I don't understand? Yes.
- Will you pardon my silly questions? There are no silly questions.

**Feel free to ask me questions and give me feedback!**

# Plans for the Second Half

course time: Tuesdays 13:20–16:10

## Instructor: Michael

- flipped classroom
- before/after class: learn with class video + “with-video” quizzes
- class time: additional learning activities, e.g., in-depth explanation of select topics, problem solving. Break as necessary.
- 2-hr class activities every 2-3 weeks, e.g., Kahoot, collaborative coding, 大地遊戲.

## Topics

- String Matching
- Linear-Time Sorting
- Disjoint Set
- Hashing
- Red Black Tree
- Graph
- B Tree

# 欣穆的碎碎念

- GOAL:  
80% of the students understand 80% of the course content.
- Tips:
  - Visualize to understand.
  - Alternate between different modes.
  - Put less emphasis on grades. Re-think your purpose.
  - Review / practice things you just learned.



- 講冷笑話的冰山學長

# Teaching Team

- email: dsa\_ta@csie.ntu.edu.tw
  - 24 TAs and 2 instructors around **with tracking**, significantly faster and more reliable than sending to individual
  - for homework questions, please tag your email (TBA) for efficient processing
- TA office hours: (TBA)
- communicate with us:
  - Questions about classes: leave questions under class video on COOL
  - Questions about homework: leave general questions on discussion forum on COOL (preferred) or tagged email
  - Announcements: sent via COOL to your NTU email. Make sure your NTU email can receive such announcements

**very friendly TAs; ask them more questions!**

# Teaching Team

吃飯組：林軒田



睡覺組：蔡欣穆



程式組：黃品硯



程式組：林瑋毅



程式組：林庭風



程式組：塗季芸



# Teaching Team

程式組：鄭豫澤



程式組：謝文傑



課活組組長：謝議霆



課活組：陳昱好



課活組：曾揚哲



課活組：徐敬能



# Teaching Team

作業組組長：劉俊緯



作業組：彭光湧



作業組：曾子顏



作業組：陳玉恆



作業組：林采鋒



作業組：吳由由



# Teaching Team

作業組：王風意



作業組：陳明信



大助教&行政組組長：裴梧鈞



行政組：余友竹



行政組：陳富中



行政組：蕭啓湘



# Teaching Team

作業組：熊育霆



作業組：陳威翰



顧問組：羅啓帆



# THE Principle

Taking any unfair advantages over other class members is not allowed.  
It is everyone's responsibility to maximize the level of fairness.

- sleeping? fine, but no snoring
- cellphone? fine, but silent mode, and speak outside
- ...

**applies to instructors, TAs, students**

# Honesty

**NO CHEATING**

**NO LYING**

**NO PLAGIARISM**

**very very very very serious consequences  
(negative scores, Fail, kicked-out-of-university, etc.)**

# Grade

- raw score is calculated from
  - homework (30%: 3x10%)
  - in-class performance and participation (20%)
  - midterm exam (15%) and final exam (20%)
  - final project (15%)
- raw score goes through some order-preserving normalization steps, **not just using default thresholds of university**
  - raw score 80 with term rank A: possible
  - raw score 80 with term rank B: possible
  - raw score 60 with term rank F: possible
  - raw scores 80, 60 with term scores B, B: possible, but unlikely
  - raw scores 80, 60 with term scores F, B: **impossible**

from the principle: **no individual score change**

# Collaboration and Open-Book

- homework discussions: encouraged
- but fairness?
  - **write the final solutions alone and understand them fully**
  - **properly cite your references for all problems**
    - Example 1: Reference: wikipedia page at [URL]
    - Example 2: Discussion with B87506055
    - Example 3: Reference: book [title] p.xx
- other documents (books, notes, Internet):  
consulted **and cited**, but **not copied from**
- no need to lend/borrow solutions

We are committed to look for **signs of plagiarism**  
(and use **tools** to do so)!

# Collaboration and Open-Book

to maximize fairness (everyone's responsibility),  
**lending/borrowing/buying/selling/trading not allowed**

to maximize fairness (everyone's responsibility),  
**lending/borrowing/buying/selling/trading not allowed**

to maximize fairness (everyone's responsibility),  
**lending/borrowing/buying/selling/trading not allowed**

Deal? If your classmate wants to borrow homework from you,  
what do you say?

# Homework

- two to four programming problems per homework (more details later)
  - graded by online judge, with a maximum of **10 submissions per day**
  - learn to **test your own code!**
  - will enforce git-based judge submission—**learn to use version control tools**
- about three non-programming problems
  - graded on Gradescope (more details later)
  - proofs, analysis
  - design of data structure/algorithm “by hand”

# Homework Submissions

- electronic submission to designated platforms (judge, gradescope) only
- no individual extension unless not violating the principle (e.g. institute-established cases of illness or emergency)

late penalty: linear increase from 0% to 100% after five days  
(e.g. lose 20% of value every day)

## Exams and Project

**TBA**

(still planning, given the uncertain class size)

# Textbook

Introduction to Algorithms, 3rd edition, 2009, MIT Press by Cormen, Leiserson, Rivest, and Stein.

- the ‘bible’ to learn about algorithms
- **learning to read a textbook** can help you deepen your knowledge
- getting the book:
  - do not use a pirated copy (**learning to respect copyright!**)
  - NTU Library: reserved copy in the shared course material area
  - R536: will put one shared copy to be read in the room
  - **If the book is not affordable to you: email me**  
**(htlin@csie.ntu.edu.tw)** privately and I'll see how I can help.

# Mandarin and English

- Mandarin: main language
- English: often encountered
  - coding, website, assignments, some teaching ...
  - important for your future and you are recommended to practice

**don't be afraid of English**

# How to Pass the Class?

COOL: <https://cool.ntu.edu.tw/courses/4621>

email: dsa\_ta@csie.ntu.edu.tw

- catch up from day 1
  - solve homework 0 this week**, if you have not done so already
- ask questions and interact frequently with teaching team
- have fun writing programs
- understand writing proof

**Enjoy the Class! Questions?**