Practical Data Structures and Algorithms
Introduction

Instructor: Chien-Yu Chen
BIME@NTU
Monday 14:20-17:10

上課方式

- 選項一：實體課程
  - 時間：星期一 789
  - 地點：知武館 R401
    - 請助教協助於14:00刷卡開門
- 選項二：課程直播
  - Google Meet
- 選項三：課程側錄
  - 最晚於每週二晚上上傳到NTU COOL

Chien-Yu Chen, BIME@NTU

課程評量

- **40%** 程式作業
  - 程式語言：Java 或 Python 二選一
- **30%** 期中考 (實體課堂考試)
  - 期中考週 (11/9)
- **30%** 期末考 (實體課堂考試)
  - 期末考週 (1/11)

This course is about

- Data structures
- Algorithms
- We will focus more on algorithms than on data structures
- Algorithms: methods for solving problems that are suited for computer implementation

Algorithms and data structures

- Most algorithms involve methods of organizing the data used in the computation.

- Objects created in this way are called data structures.

- Thus, algorithms and data structures go hand in hand.

- The efficiency of algorithms can be largely improved after considering a different way of storing data.

Chien-Yu Chen, BIME@NTU

Problem space and solution space

- A problem can have more than one solutions.

- We call the solution an algorithm here, which consists of a number of procedures to achieve the task.

- An algorithm must be correct.

- One algorithm might be faster or much faster than the others.

Chien-Yu Chen, BIME@NTU

An example of problems

Figure 1.2
A large connectivity example

The objects in a connectivity problem might represent connection points, and the pairs might be connections between them, as indicated in this idealized example that might represent wires connecting buildings in a city or components on a computer chip. This graphical representation makes it possible for a human to spot nodes that are not connected, but the algorithm has to work with only the pairs of integers that it is given. Are the two nodes marked with the large black dots connected?
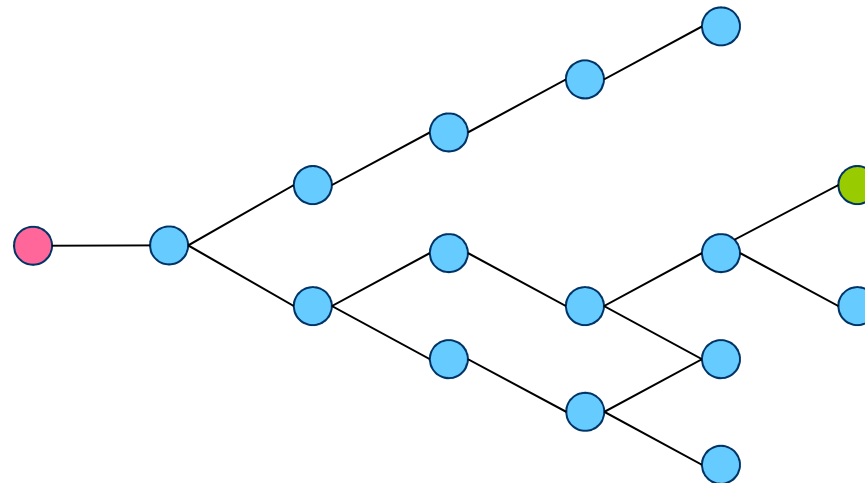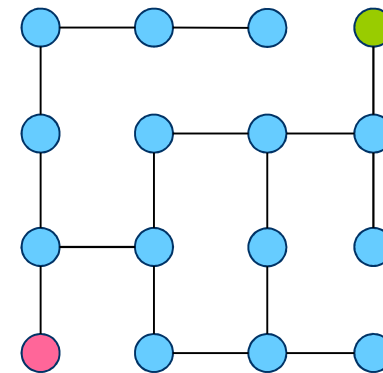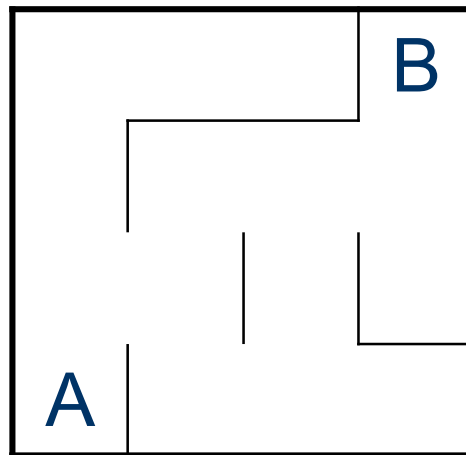
Chien-Yu Chen, BIME@NTU
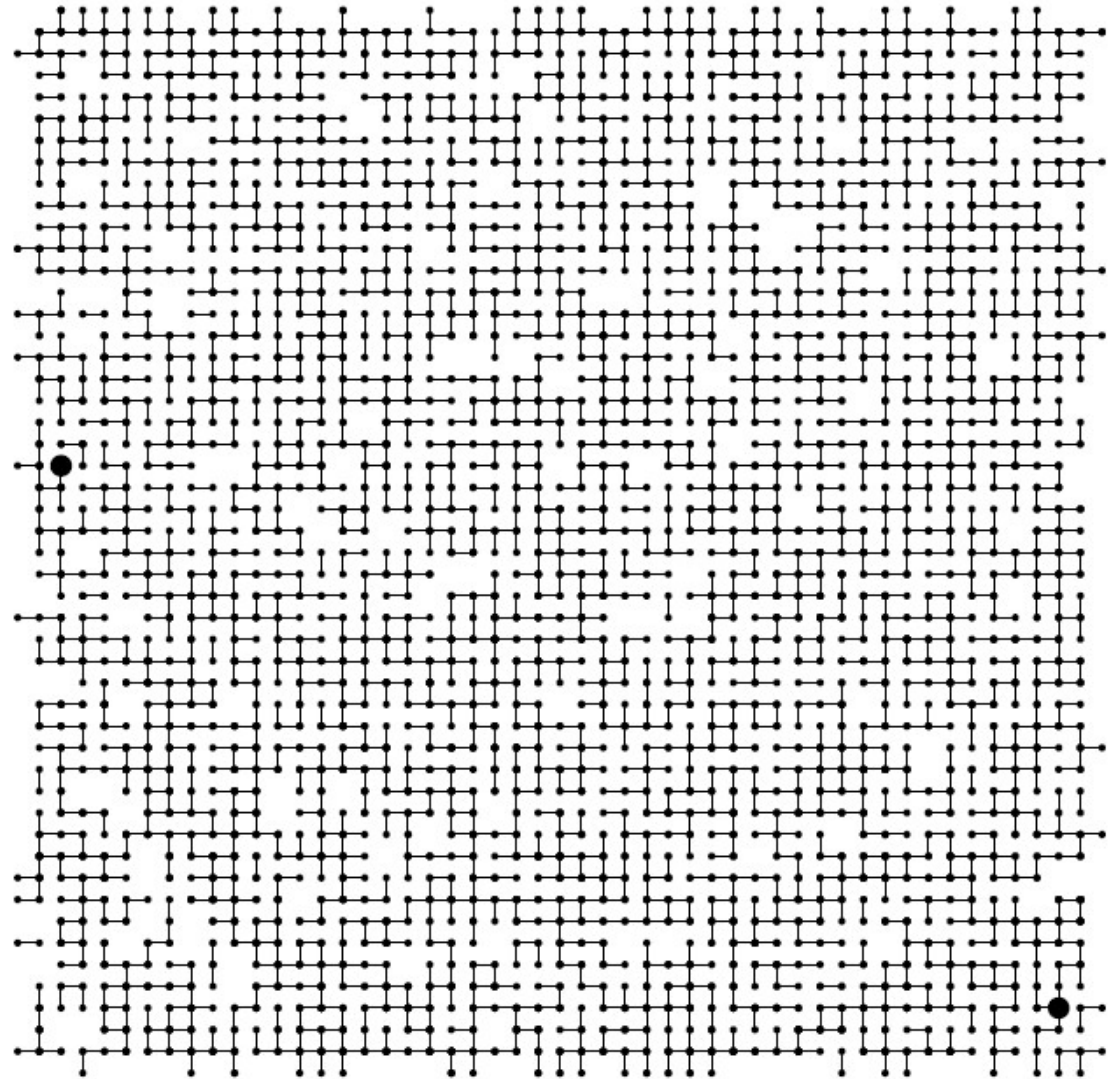
Difficulty of the problems

- P1: is p and q connected?
- P2: find all the paths from p to q
- P3: find the shortest path that goes from p to q

What you will learn and will not learn in this course

Textbooks

1. **Algorithms** 4th Edition

   by Robert Sedgewick and Kevin Wayne

   (http://algs4.cs.princeton.edu/home/)

2. **Introduction to Programming in Java**

   by Robert Sedgewick and Kevin Wayne

   (http://introcs.cs.princeton.edu/java/home/)

3. **Introduction to Programming in Python**

   by Robert Sedgewick , Kevin Wayne, and Robert Dondero
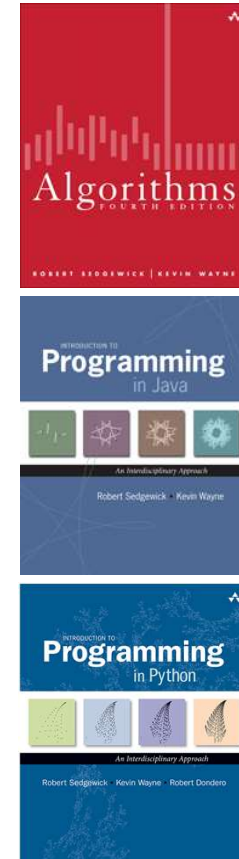
   (https://introcs.cs.princeton.edu/python/home/)

Chien-Yu Chen, BIME@NTU

https://www.coursera.org/learn/algorithms-part1/home/welcome

Chien-Yu Chen, BIME@NTU

After today's class

- You should
  - Register an account in Coursera
    - https://www.coursera.org/
  - Enroll the course: Algorithms, Part I
    - https://www.coursera.org/learn/algorithms-part1
  - Listen to the lecture: Course Introduction
    - https://www.coursera.org/learn/algorithms-part1/lecture/buZPh/course-introduction
  - Prepare programming environment
    - Suggested: IntelliJ
      https://lift.cs.princeton.edu/java/windows/
  - Learn Java or Python online
    - Example: https://www.learnjavaonline.org/
    - Example: https://www.learnpython.org/

Chien-Yu Chen, BIME@NTU

Time table

- 9/14 (一) HW1 announced in NTU COOL
- 9/25 (五) HW1 due (21:00)
- 9/26 (六) 網路加選課程截止（下午6時截止）
- 9/27 (日) 網路退選課程截止（系統開放至28日上午8時）

（上述加退選截止時間仍以台大選課網之公告為準）

- Online TA hour: TBA

Chien-Yu Chen, BIME@NTU

FB 社團

- https://www.facebook.com/groups/2162064870594047

NTU-1091-PDSA
🔒 私密社團 · 36 位成員

批改系統

- 網址：TBA
- 申請帳號
  - 請與助教聯繫(NTU COOL上有助教的email)
  - 僅開放給修課同學

- 注意：
  - 勿將自己的程式碼與其他同學分享
  - 勿直接使用網路上找到的程式碼
  - 如果A同學與B同學的程式碼一模一樣，兩位同學的該次作業都以0分計算

Chien-Yu Chen, BIME@NTU

Randy Pausch

- ## The Last Lecture: Really Achieving Your Childhood Dreams

https://www.youtube.com/watch?v=V22yKlUt4F4

"The brick walls are there for a reason. The brick walls are not there to keep us out. The brick walls are there to give us a chance to show how badly we want something. Because the brick walls are there to stop the people who don't want it badly enough. They're there to stop the other people."
— Randy Pausch, *The Last Lecture*

Chien-Yu Chen, BIME@NTU