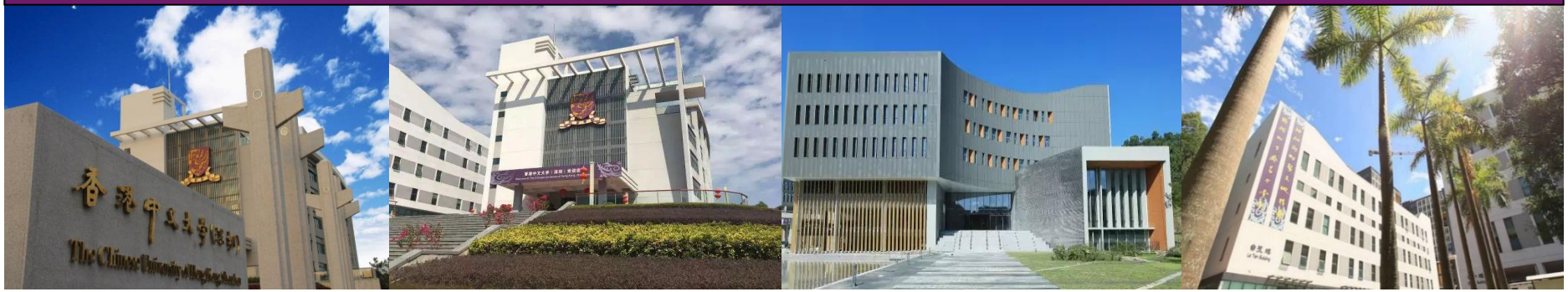
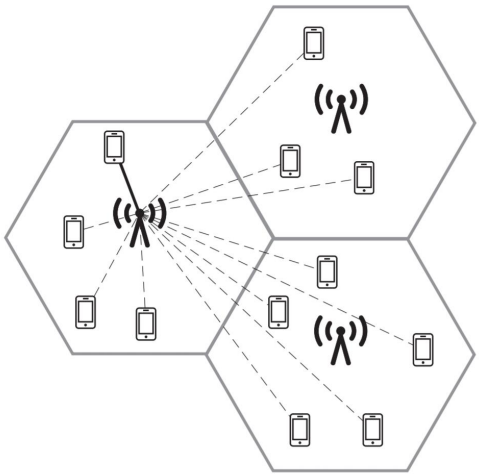
CSC3100: Introduction

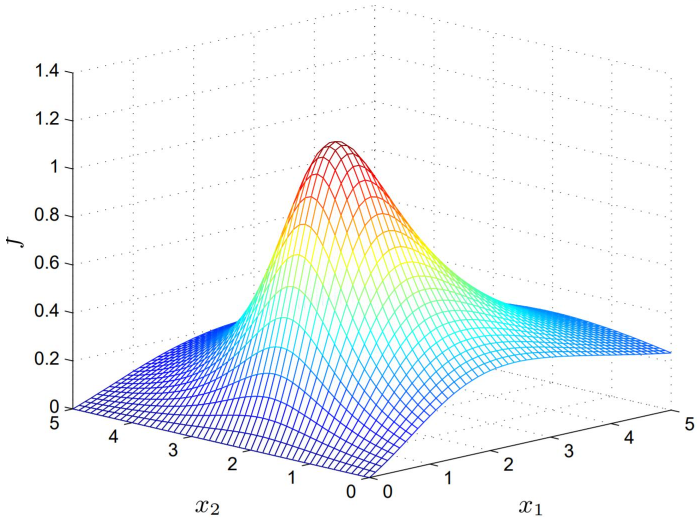
Kaiming Shen

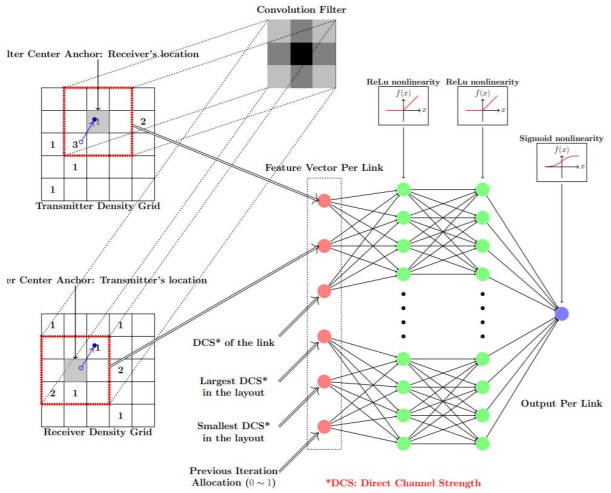


Instructor

- Dr. Kaiming Shen
- Tenure-track assistant professor with SSE
- Email: shenkaiming@cuhk.edu.cn
- https://sseold.cuhk.edu.cn/en/content/13693
- 5G, optimization for big data, machine learning
  Intern positions available







Lecture Schedule

- Session 1
  - 1:30-3:30 pm, Monday
  - 1:30-2:30 pm, Wednesday
- Session 2
  - 3:30-5:30 pm, Monday
  - 3:30-4:30 pm, Wednesday
- Office hours: TBD
- Attend either session
- Classroom: Cheng Dao Bldg 101

TA

Name	Email
王钦	219019047@link.cuhk.edu.cn
王星超	217019013@link.cuhk.edu.cn
薛凯文	217019024@link.cuhk.edu.cn
吕虹晔	218012026@link.cuhk.edu.cn
邱陵腾	qiulingteng@stu.hit.edu.cn
段海涵	duanhaihan@stu.scu.edu.cn

Marking

Tutorial Schedule

- Mon (7-9pm), Tue (6-9pm), Wed (6-9pm)
- Classroom: Room 210, Teaching C Bldg

18 00		CSC3100-Tutorial-D1/03 Classroom:TC_210 Week:1-14	CSC3100-Tutorial-D1/06 Classroom:TC_210 Week:1-14
19 00	CSC3100-Tutorial-D1/01 Classroom:TC_210 Week:1-14	CSC3100-Tutorial-D1/04 Classroom:TC_210 Week:1-14	CSC3100-Tutorial-D1/07 Classroom:TC_210 Week:1-14
20 00	CSC3100-Tutorial-D1/02 Classroom:TC_210 Week:1-14	CSC3100-Tutorial-D1/05 Classroom:TC_210 Week:1-14	CSC3100-Tutorial-D1/08 Classroom:TC_210 Week:1-14

Grading

- 4 Assignments (written+coding): 40%
  - Written: pseducode
  - coding: Java
- Midterm (written): 20%
- Final Exam (written): 40%

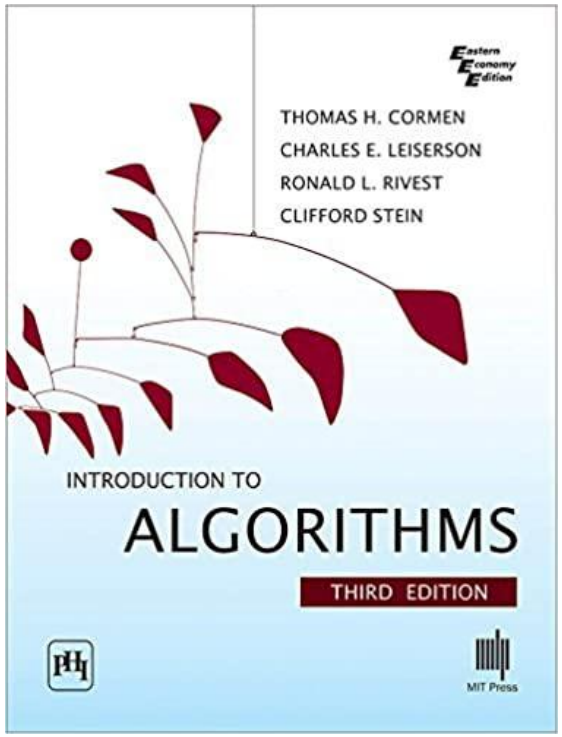
Study Groups

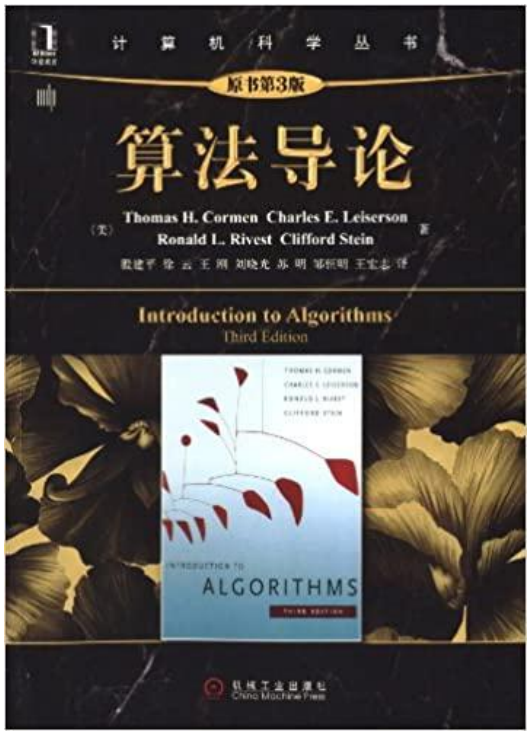
- Students can form study groups of up to 3 people to discuss lectures and assignments

- But each member must reconstruct answers indepdently

Textbook

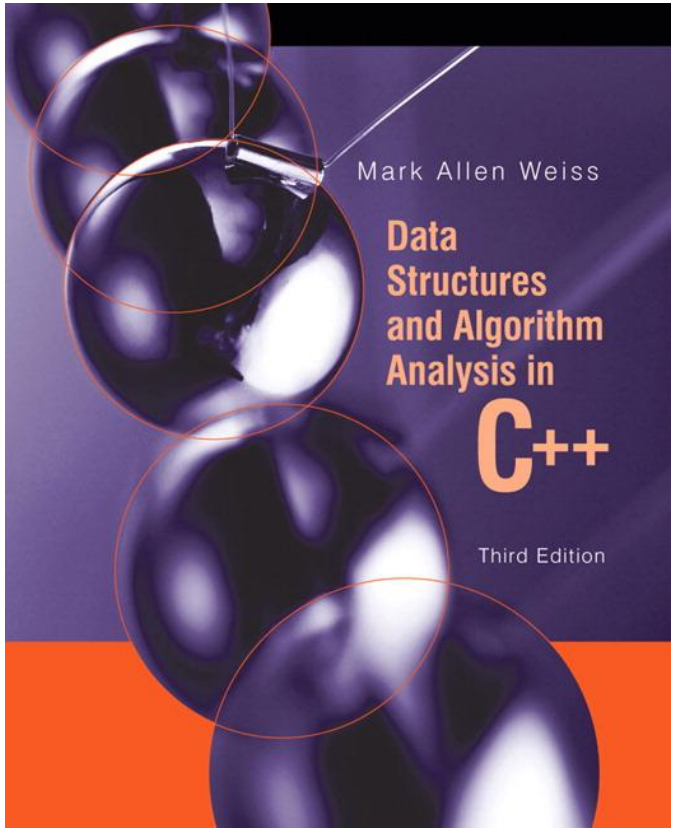
- "Introduction to Algorithms (3rd Edition)", a.k.a. "CLRS"
- The MIT Press, 2009





Reference

- "Data Structures and Algorithm Analysis in C++" by Mark Allen Weiss



What is Data Structure?

A data structure is a way to store and organize data in order to facilitate access and modifications.

- Validate one Chinese ID in 1.4 billion people
- Find the best driving route

Efficiency matters!

Pseudocode

- Loop: **for**, **while**, **repeat-until**
- Condition: **if-else**, **break**, **continue**

```
for i = 1 to 5
    if i==3
        break
    print i
```

Output: 1 2

```
for i = 1 to 5
    if i==3
        continue
    print i
```

Output: 1 2 4 5

- Assignment: i=a versus i==a
- Boolean algebra: **and**, **or**

A	B	A **AND** B
0	0	0
0	1	0
1	0	0
1	1	1

A	B	A **OR** B
0	0	0
0	1	1
1	0	1
1	1	1

Pseudocode

- Array: A[1..j] = A[1], A[2], …, A[j]

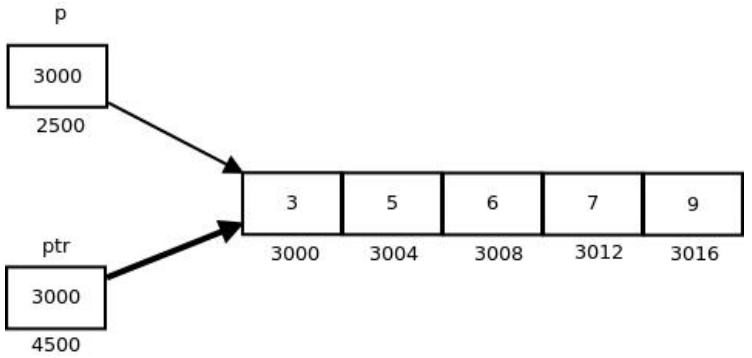
  A = (1,2,3)

  B = A

  A[1] = 4

  print B[1..3]

  output: (4,2,3)



- object & attribute: A.length
- Passing argument by value/object:

x=1	A.length=1
f(x) {x=2}	f(A) {A.length=2}
print x output: 1	print A.length output: 2

Exponents

$$X^A X^B = X^{A+B}$$

$$\frac{X^A}{X^B} = X^{A-B}$$

$$(X^A)^B = X^{AB}$$

$$X^N + X^N = 2X^N \neq X^{2N}$$

$$2^N + 2^N = 2^{N+1}$$

Logarithm

$$X^A = B \text{ if and only if } \log_x B = A$$

– *Useful equalities*

$$\log_A B = \frac{\log_c B}{\log_c A}; C > 0$$

$$\log AB = \log A + \log B$$

$$\log\left(A/B\right) = \log A - \log B$$

$$\log x < x, \quad \forall x > 0$$

$$\log 1 = 0, \log 2 = 1, \log 1024 = 10, \log 65536 = 16$$

Note: All log are to be base 2 in this course unless specified otherwise.

Arithmetic Sequences and Series

The *n*th Term of an Arithmetic Sequence

The *n*th term of an arithmetic sequence $a, a + d, a + 2d, \ldots$ is given by

$$a_n = a + (n - 1)d$$

Formulas for the Sum of an Arithmetic Series

$$S_n = \frac{n}{2}[2a + (n - 1)d]$$

$$S_n = n\left(\frac{a + a_n}{2}\right)$$

Geometric Sequences and Series

*n*th Term of a Geometric Sequence

The nth term of the geometric sequence, $a, ar, ar^2, \ldots$ is given by

$$a_n = ar^{n-1}$$

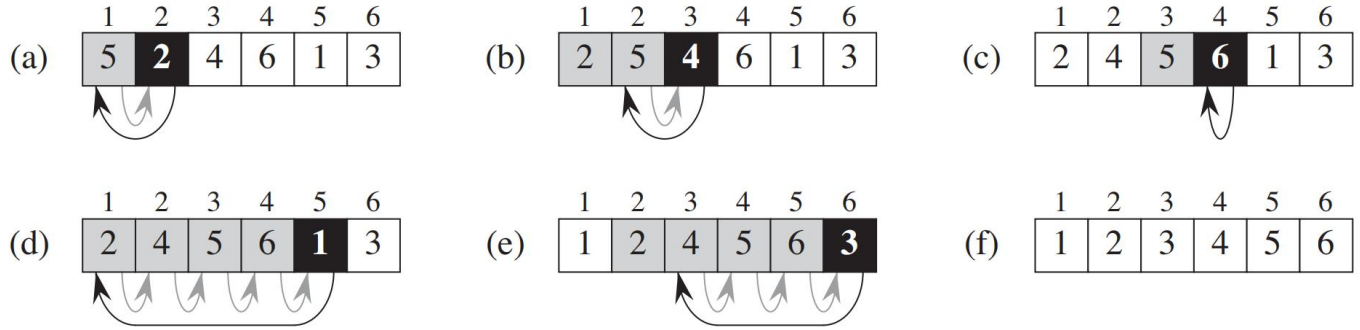
Formula for the Sum of a Geometric Series

Let S_n denote the sum $a + ar + ar^2 + \cdots + ar^{n-1}$, and assume that $r \neq 1$. Then

$$S_n = \frac{a(1 - r^n)}{1 - r}$$

Data Sorting

Q: Sort *A*=(5, 2, 4, 6, 1, 3) in ascending order.



INSERTION-SORT(A)

```
1  for j = 2 to A.length
2      key = A[j]
3      // Insert A[j] into the sorted sequence A[1 .. j − 1].
4      i = j − 1
5      while i > 0 and A[i] > key
6          A[i + 1] = A[i]
7          i = i − 1
8      A[i + 1] = key
```

Data Sorting

- However, if the group consists of 10^7 numbers, it takes the insertion method more than 2 days to complete sorting.

- In contrast, by using data structure wisely, we can cut down time to less than 30min.

Data Sorting

Q: Given a group of N numbers, determine the k^{th} largest, where k<=N.

Solution 1:

1) read N number into an array,
2) sort the array in descending order,
3) return the element in position k.

Data Sorting

Solution 2:

1) read the first k elements into an array and sort them in descending order,
2) each remaining element c is read one by one,
   2.1) it is ignored if it is smaller than or equal to the k^{th} element in the array
   2.2) otherwise, it is placed in its correct spot in the array, bumping one element out of the array.
3) the element in the k^{th} position is returned as the answer.

Data Sorting

- Which solution is better ?
  - By simulation
  - By theoretical analysis
- Is either algorithm good enough (particularly when N is very large)?
  - A simulation using 1 million elements and k = 500,000 will show that NEITHER algorithm finishes in a reasonable amount of time
  - Is there a better algorithm?

Summary

Throughout this course we will consider

- How to estimate the running time of a program for large inputs
- How to compare the running times of two programs
- How to improve the speed of a program, and to determine the program bottlenecks.