香港中文大學（深圳）
The Chinese University of Hong Kong

CSC3100 Data Structures
Course information

Yixiang Fang
School of Data Science (SDS)
The Chinese University of Hong Kong, Shenzhen

About instructor (Yixiang Fang)

- Background
  - PhD: 2013-2017, HKU
  - Postdoc: 2017-2018, HKU
  - Postdoc: 2018-2020, UNSW
  - Associate Professor: 2021-present, CUHK-SZ
- Contact
  - Email: fangyixiang@cuhk.edu.cn
  - Office: Room 417d, Dao Yuan Building
- Research
  - General area: querying, mining, and analytics of big data
  - Topics: graph queries, graph mining, graph learning, etc.

Lecture location and time

- Three sessions
  - Leading instructor: Yixiang Fang

CSC 3100 - Data Structures

View All Sections First 1-3 of 11 Last

Section	L01-LEC(3192)	**Enrollment Total**	157	**Status**
Session	Regular	**Enrollment Capacity**	180	the remaining quota is reserved

Days & Times	Room	Instructor	Meeting Dates
MoWe 10:30 - 11:50	Teaching B Bldg 202	Yixiang Fang	2022/09/05 - 2022/12/16

Section	L02-LEC(3193)	**Enrollment Total**	90	**Status**
Session	Regular	**Enrollment Capacity**	180	the remaining quota is reserved

Days & Times	Room	Instructor	Meeting Dates
MoWe 15:30 - 16:50	Teaching B Bldg 103	Benyou WANG	2022/09/05 - 2022/12/16

Section	L03-LEC(3194)	**Enrollment Total**	85	**Status**
Session	Regular	**Enrollment Capacity**	180	the remaining quota is reserved

Days & Times	Room	Instructor	Meeting Dates
TuTh 14:00 - 15:20	Administration Bldg E101	Zhizheng WU	2022/09/05 - 2022/12/16

Prof. Zhizheng Wu

Prof. Benyou Wang

Lecture location and time

- Session by Yixiang Fang
  - Lecture location: Room 202, Teaching B Building
  - Lecture time (3 hours/week)
    - 10:30 am ~ 11:50 am on Monday
    - 10:30 am ~ 11:50 am on Wednesday

A break of 5~10 mins

  - Mix-mode teaching
    - Zoom ID: 671 987 8160 (No password)
    - The videos are available online (see Blackboard)
  - Office hour
    - Time: 4:30 pm ~ 5:30 pm on every Thursday
    - Location: Room 417d, Dao Yuan Building
    - Online: 671 987 8160 (No password)

TA information

- 10 TAs (5 PGTAs + 5 USTFs)
  - Leading TA: Mr. Yingli Zhou
  - Host tutorials; mark assignments & exams; office hour

Office hours of TAs

ID	Name	ZOOM-ID	PGTA/USTF	Room	Office hour	Email address
1	Yingli Zhou (leading TA)	322 497 0842	PGTA	Dao Yuan Building 2th Floor, Room 224	Fri 16:00 - 17:00	yinglizhou@link.cuhk.edu.cn
2	Junyi Ao	435 279 9601	PGTA	H.L. Tu Building, RB311	Wed 16:30 - 17:30	junyiao1@link.cuhk.edu.cn
3	Xi Chen	994 444 9359	PGTA	The bigger room of SDS Research Lab (4th Floor, Zhi Xin Building) , seat no. 100	Wed 16:00 - 17:00	xichen7@link.cuhk.edu.cn
4	Xueyao Zhang	670 422 4188	PGTA	The bigger room of SDS Research Lab (4th Floor, Zhi Xin Building), seat no. 78	Thu 16:00 - 17:00	xueyaozhang@link.cuhk.edu.cn
5	Yaomin Wang	374 048 9343	PGTA	The bigger room of SDS Research Lab (4th Floor, Zhi Xin Building), seat no. 110	Thu 15:00 - 16:00	222042013@link.cuhk.edu.cn
6	Biaolin Wen	509 139 1757	USTF	TBA	Thu 14:00 - 15:00	biaolinwen@link.cuhk.edu.cn
7	Jie Pan	437 883 9639	USTF	TBA	Mon 16:00 - 17:00	121090434@link.cuhk.edu.can
8	Qingshuo Guo	827 451 5237	USTF	TBA	Wed 14:00 – 15:00	121090151@link.cuhk.edu.cn
9	Derong JIn	809 737 4794	USTF	TBA	Thu 19:00 – 20:00	120090562@link.cuhk.edu.cn
10	Tianci Hou	447 624 4293	USTF	TBA	Wed 19:00 - 20:00	121090184@link.cuhk.edu.cn

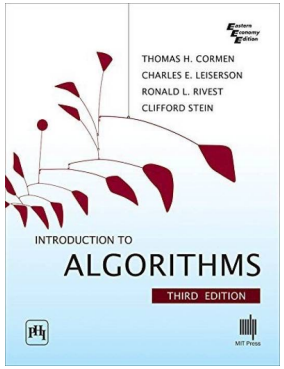
Tutorials

- Tutorials in mix-mode
  - The zoom IDs are the same as those of TAs' office hours in the previous slide
  - Check announcements in your e-mail box or Blackboard
  - Check Wechat group messages

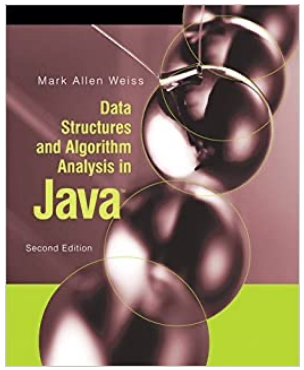
Tutorial time	Monday	Tuesday	Wednesday	Thursday	Friday
18:00 - 18:50		TA111, TA204, TA205	TA111, TA204		
19:00 - 19:50	TA111, TA204		TA111	TA111, TA205, TA204	TA111

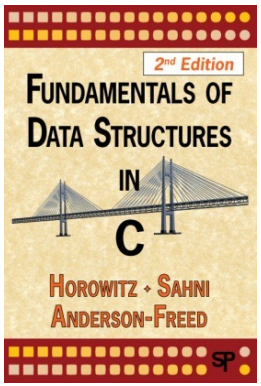
Textbook and references

- Textbook
  - T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein, Introduction to Algorithms, 3rd Edition, The MIT Press, 2009.
  - https://edutechlearners.com/download/Introduction_to_algorithms-3rd%20Edition.pdf



- References
  - M.A. Weiss, Data Structures and Algorithm Analysis in Java, 2nd Edition, Addison-Wesley, 2007.
  - Ellis Horowitz, Sartaj Sahni, Susan Anderson-Freed, Fundamentals of Data Structures in C.





Teaching methods

- Lectures (3 hours/week)
  - Important materials from the textbook will be covered
  - Please ensure you stay up with the materials
  - Java will be used for illustrating the data structures
  - Feel free to interrupt to ask questions

- Tutorials (1 hour/week)
  - Illustrate more on the difficult parts
  - Show students more examples
  - Discuss assignment/exercise
  - Answer students' questions

How to do well in this course?

- Common suggestions
  - Slides of lectures/tutorials will be uploaded to Blackboard before lectures/tutorials; learn them in advance
  - Use examples to facilitate learning
  - Data structures are not just reading materials; you need to write more codes!!!
- Special suggestions
  - If you feel difficult,
    - Try to focus on the content of slides and seek for help (TAs and instructors are ready to answer your questions!)
  - If you feel easy,
    - Read more details (e.g., theoretical analysis) in the textbook
    - Use the learned techniques to solve ACM-ICPC problems

Learn more and practice more

- ACM-ICPC
  - ACM: Association for Computing Machinery
  - ICPC: International Collegiate Programming Contest

- Online Judge (OJ) systems
  - CUHKSZ OJ: 10.26.200.13 (access in campus)
  - SJTU OJ: https://acm.sjtu.edu.cn/OnlineJudge/
  - PKU OJ: http://poj.org/

- CUHK-SZ programming team
  - Coaches: Yixiang Fang, Pinjia He
  - Campus programming contest: Oct or Nov, 2022 (one or two weeks after midterm exam)

Course assessment

Homework 39% + Midterm exam 20% + Final exam 40% + CTE 1%

	Release date	Due date	Weight	
Written assignment 1	Sep 23	Oct 7	9%	
Programming assignment 1	Oct 21	Nov 4	10%	Python/C/C++/ Java are ok
Midterm exam	Oct 29	--	20%	
Written assignment 2	Nov 11	Nov 25	10%	
Programming assignment 2	Dec 2	Dec 16	10%	Python/C/C++/ Java are ok
Final exam	Dec 17 - 24		40%	

*The above table is subject to change as the instructors see fit.

Course policy

- Your work MUST be your own
  - Cheating is against "fair-play" and will not be tolerated under any circumstances

- Assignments
  - Penalty will be imposed on copying assignments; minimum penalty is zero mark for the assignments
  - There will be penalty for late submission:
    - 0~24 hours after deadline: final score = your score x 0.8
    - 24~72 hours after deadline: final score = your score x 0.5
    - 72+ hours after deadline: final score = your score x 0

Students with diverse backgrounds

Year	Percentage
1st year	0
2nd year	45.2%
3rd year	50.3%
4th year	4.5%
Overall	100%

- Student distribution (157)
  - SME (18): 11.5%
    - Financial engineering: 18
  - SSE (24): 15.3%
    - Electronic Info Engineering: 20
    - Mathematics and Applied Mathem: 4
  - SDS (115): 73.2%
    - Data Science and Big Data Tech: 90
    - Computer Science and Engineering: 25

Please try to consider your classmates before complaining

Feedback & acknowledgements

- Feedback is important and also welcome!
  - Talk to course instructors and TAs, or send us emails
  - Please talk to us before complaining to others
  - Questionnaire

- Acknowledgements
  - Lecture slides
    - Prof. Xiang Wan (CUHK-SZ), Prof. Kaiming Shen (CUHK-SZ), Prof. Wenye Li (CUHK-SZ), Prof. Minming Li (CityU, HK), Prof. Zengfeng Huang (FDU), Prof. Jane You (PolyU, HK), Prof. Sibo Wang (CUHK)
  - Tutorial slides
    - Mr. Xingchao Wang (CUHK-SZ), Mr. Panwen Hu (CUHK-SZ), Mr. Ziteng Weng (CUHK-SZ)

Extra bonuses

- Extra bonuses for students who perform very well
  - Recommend industry internship in companies like BATH
  - Offer research assistant (RA) positions and you may have the chances to publish academic papers!
  - High chances to be PhD students in my research team
  - Write recommendation letters for your Master or PhD applications

香港中文大學（深圳）
The Chinese University of Hong Kong

CSC3100 Data Structures
Lecture 1: Introduction

Yixiang Fang
School of Data Science (SDS)
The Chinese University of Hong Kong, Shenzhen

Outline

- Introduction
  - Why take this course?
  - Basic concepts, e.g., abstract data type (ADT)
  - Relationship of ADT, data structures, and algorithms
  - Topics in this course
  - Tentative teaching plan

Why take this course?

- Required course
  - Also very important and very useful
  - A fundamental course in computer science
- Learn to save the data (data structure) and manipulate the data (algorithms) effectively and efficiently
- No single data structure fits all scenarios
  - Array: friendly to search (if sorted), not friendly to updates
  - List: friendly to updates, not friendly to search
- How useful in practice?
  - E.g., validate one Chinese ID in 1.4 billion people
  - E.g., find the best driving route
  - E.g., find webpages in Google

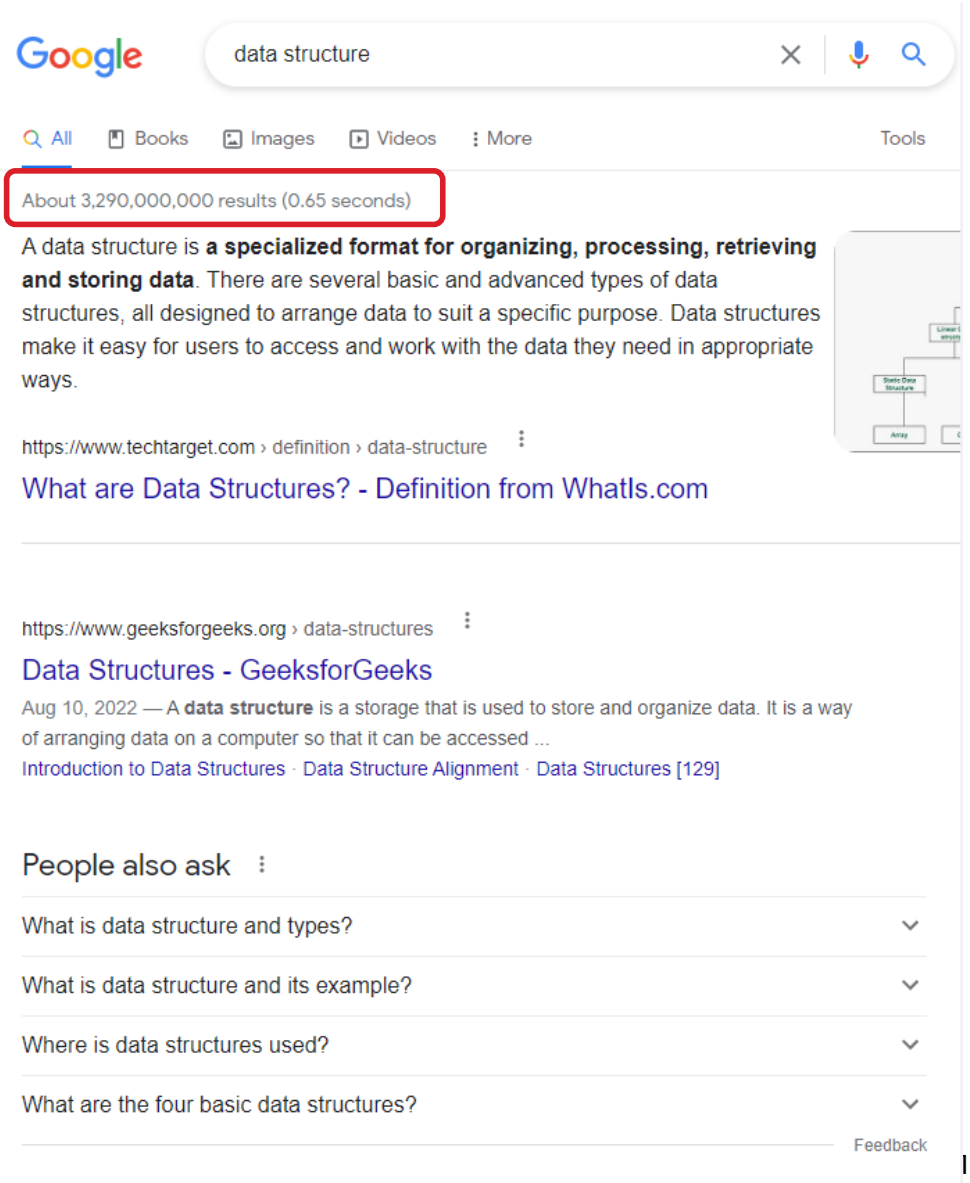
Real examples

- Route planning
  - Find the shortest path between two specific locations
  - Input:
    - A road network
    - A source node (location)
    - A destination node (location)
  - Output
    - A path, or a sequence of edges, with the shortest total distance

Real examples

- Google search
  - Find the documents matching your query keywords
  - Use sophisticated algorithms to create an index structure which is just a data structure

30 trillion (30×10^{12}) webpages



One sentence about this course

This course is about how to use computing resources and tools to solve practical problems **correctly** and **efficiently**

Logical thinking and math

Good at data structures, algorithms, and programming

Data abstraction

- A clear separation between
  - the abstract properties of a data type and
  - the concrete details of its implementation

- Example: smartphone
  - Users do not know
    - How calls are made (e.g., CPU, memory, battery, and electric wire)
    - How the phone accesses the Internet
    - How the data is stored in the phone
  - Users do know
    - To make a call: input a person's phone number
    - To access the Internet: open the browser

Abstract Data Type (ADT)

- An ADT is for encapsulation (information hiding)
  - The implementation of an ADT and its operations can be localized to one section of the program
  - Procedures that make use of the ADT can safely ignore its implementation details

- Benefit of ADT
  - User-friendly: users do not need to know the mechanisms of how to connect to the Internet
  - Designer-friendly: designer can change mechanisms without affecting users
  - Protection: others cannot know your secrets!

How to separate?

- An ADT only provides the definition of operations
  - It consists of names of every operation (function), the type of its arguments, and the type of its result

```
1  ADT IntegerSet
2   IntegerSet createEmptySet();
3   IntegerSet addElementToSet(integer, SetA);
4   Boolean search(integer, SetA);
5   IntegerSet intersection(setA, setB);
6   IntegerSet union(setA, setB);
7   IntegerSet difference(setA, setB);
```

  - Does not reveal the internal implementation details
    - How the set is represented? Array, List, Tree, etc.?
    - How the operations are implemented? Many different ways

Relationships: ADT, DS, algorithms

- A data structure (DS): implementation of an ADT
  - List ADT
    - Implementation: ArrayList or LinkedList
  - Set ADT
    - Implementation: Hash Table or Red-Black Tree

- An algorithm: implementation of operations in ADT

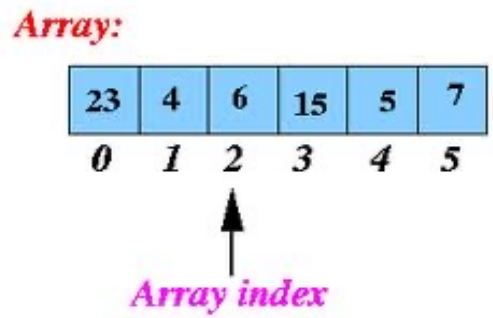
Input
A sequence of numbers: $a_1, a_2, \ldots a_n$

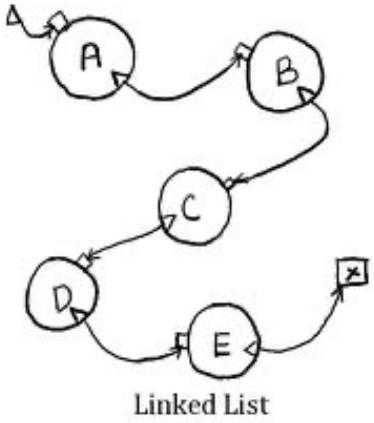
$\rightarrow$ Sorting Algorithm $\rightarrow$

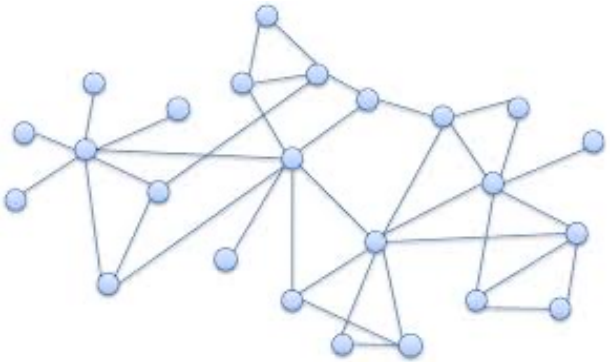
Output
A permutation (reordering) $\langle a_1', a_2', \ldots a_n' \rangle$ of the input sequence, $a_1' \leq a_2' \leq \ldots \leq a_n'$

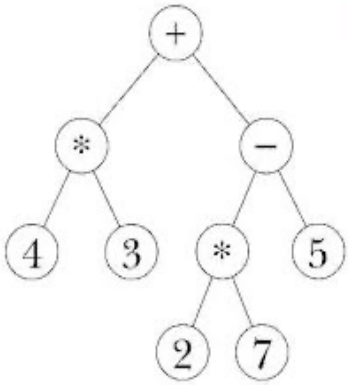
What are the common ADTs?

- *Arrays, lists, stacks, queues, trees, graphs,* etc.









Topics covered

- How to define "efficient"?
  - Time complexity analysis
- No single data structure fits all scenarios
  - Array: Efficient for sorting and search, not for updates
  - List: Efficient for updates, but not in searching/sorting
  - Stack/queues: Fast for certain types of updates, not in searching/sorting
  - Trees: Heap & binary search trees
  - Hash tables: Hash functions
  - Graphs: Graph algorithms
- Efficient operations
  - Sorting
  - Searching/Updates on different data structures (Discussed when learning the specific data structure)

Tentative teaching plan

*This tentative teaching plan is subject to change as the instructor sees fit.

Week	Content
1	Course overview, Java basics
2	Arrays, insertion/merge sort
3	Time complexity
4	List
5	Stack, queues
6	More sorting algorithms
7	Trees

Week	Content
8	Midterm exam
9	Trees
10	Hashing
11	Graphs
12	Graphs
13	Graphs, other data structures
14	Course review

Throughout this course, we will see:
- How to estimate the time cost of a program
- How to select proper data structure(s) to solve real problems
- Techniques to improve the speed of a program

Example: selection problem

- Problem
  - Given N numbers, determine the k^{th} largest, where N > k

- Solution 1:
  1) read N number into an array
  2) sort the array in descending order
  3) return the element in position k

- Solution 2:
  1) read the first k elements into an array and sort them in descending order
  2) each remaining element 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

Example: selection problem

- Two natural questions:
  - 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?
- What is the conclusion?
  - Writing a working program is not good enough!
  - The optimum solution: a correct algorithm with minimum resources and minimum running time

Exercise

- Is it possible to find a better algorithm to select the k^{th} largest number?

Mathematics review

- Exponents

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

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

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

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

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

Mathematics review

- Logarithm definition:

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

- All log are to be base 2 unless specified otherwise

– *Useful equalities*

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

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

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

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

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

Mathematics review

- Series: arithmetic series

$$\sum_{i=1}^{N} i = \frac{N(N+1)}{2}$$

$$\sum_{i=1}^{N} i^2 = \frac{N(N+1)(2N+1)}{6}$$

Example: To find the sum

2+5+8+….+ (3k-1)

= 3(1+2+3+…+k)

- (1+1+1+….+1)

$$= \frac{3k(k+1)}{2} - k$$

Mathematics review

- Series: geometric series

$$\sum_{i=0}^{N} A^i = \frac{1 - A^{N+1}}{1 - A}$$

$$If\ 0 < A < 1, then$$

$$\sum_{i=0}^{N} A^i \leq \frac{1}{1 - A}$$

- Derivation

Let $S = 1+A+A^2+\ldots\ldots$ (1)

where, $0<A<1$

then $AS = A+A^2+A^3+\ldots$(2)

Subtracting (1) and (2), we get $S-AS <= 1$, i.e.

$$S \leq \frac{1}{1 - A}$$

Mathematics review

- To prove a false statement:
  - proof by counter example

  e.g., Fibonacci number:

  $F_0 = 1, F_1 = 1, F_{k+1} = F_k + F_{k-1}$

  To show the statement $F_k \leq k^2$ is false, we can compute a concrete counter example, e.g.,

  $F_{11} = 144 > 11^2$.

- To prove a correct statement
  - proof by induction
    - (1) proving a base case
    - (2) inductive hypothesis
  - proof by contradiction
  - (1) assume it is false
  - (2) show that this assumption is false