Data Structure

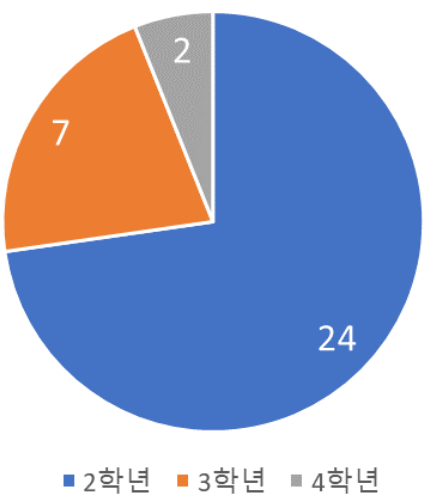
# Course Overview

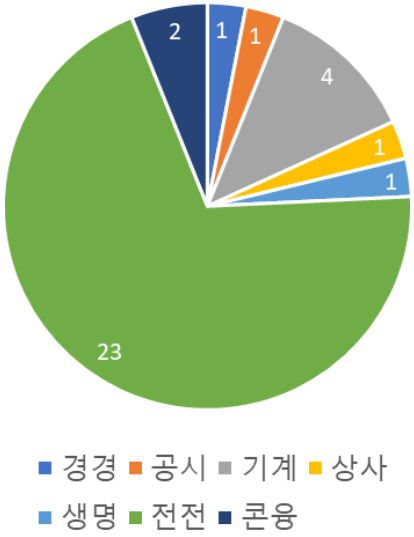
Shin Hong

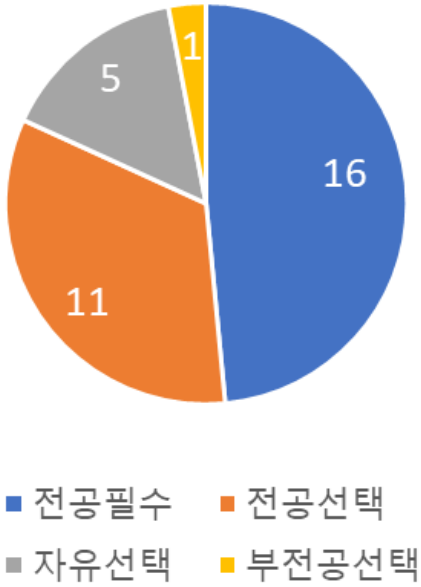
Mar 3, 2020

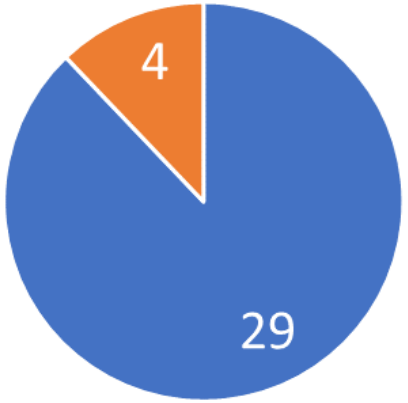
# Class





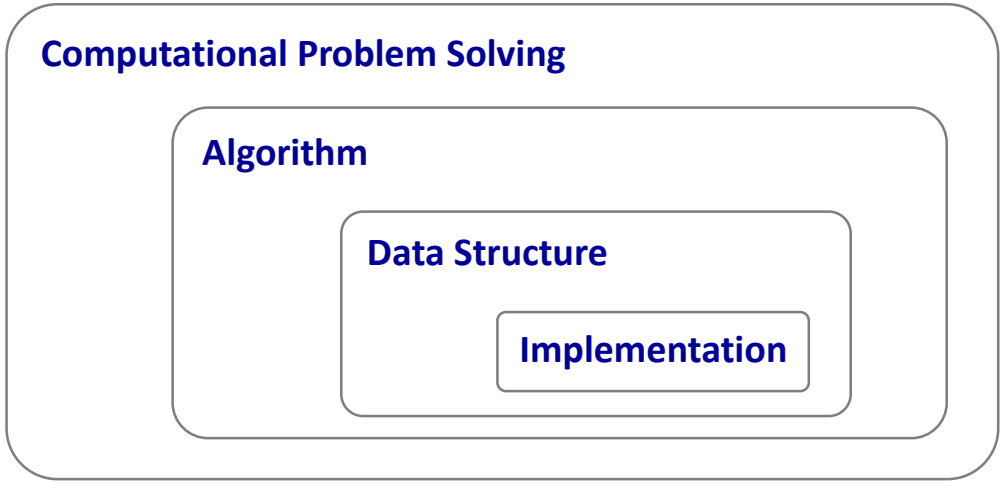




- 33 students
- 1 instructor: Shin Hong
  - hongshin@handong.edu / http://hongshin.github.io / OH 313
- 3+ teaching assistants
  - Hyerin Leem hyerinleem@handong.edu (coordinating TA)

# Course Objectives



- Understand the principles of fundamental algorithms
  - study frequent structures of computational problems
  - understand key ideas in scientific solutions
  - study each set of algorithms related with a certain data structure
- Build essential background/skills for pursing a degree in CS
  - understanding foundational concepts/ideas of computer science
  - learning how to read technical description
  - practice writing computational solutions in programming languages



# Course Materials



- Course website: http://github.com/hongshin/DataStructure/wiki

- Textbooks

  1. **Fundamentals of Data Structures in C** by Horowitz et al.

  2. **Data Structure and Algorithm Analysis, C++ version of Edition 3.2** by Clifford A. Shaffer
     http://people.cs.vt.edu/~shaffer/Book/

- Programming Labs
  - There will be 8 to 10 programming lab sessions at the Friday meeting
  - In each lab session, you are given 2 or 3 programming problems to solve within the lab session
  - Two students will work together at each lab session
    - basically you will have a new partner in every lab session

# Topics and Schedule (tentative)



| Date | Topic |
|---|---|
| Mar 3 , Tue | course introduction |
| Mar 6, Fri | linked list |
| Mar 10, Tue | stack |
| Mar 13, Fri | queue |
| Mar 17, Tue | doubly-linked list |
| Mar 20, Fri | tree (1/2) |
| Mar 24, Tue | tree (2/2) |
| Mar 27, Fri | heap |
| Mar 31, Tue | disjoint set |
| Apr 3, Fri | * lab 1 |
| Apr 7, Tue | graph (1/3) |
| Apr 10, Fri | * lab 2 |
| Apr 14, Tue | graph (2/3) |
| Apr 17, Fri | * lab 3 |
| Apr 21, Tue | graph (3/3) |
| Apr 24, Fri | ** midterm exam: 4-6 PM |

| Date | Topic |
|---|---|
| Apr 28, Tue | sorting (1/3) |
| May 1, Fri | * lab 4 |
| May 5, Tue | ** national holiday |
| May 8, Fri | sorting (2/3) |
| May 12, Tue | sorting (3/3) |
| May 15, Fri | * lab 5 |
| May 19, Tue | hashing |
| May 22, Fri | * lab 6 |
| May 26, Tue | priority queue |
| May 29, Fri | * lab 7 |
| June 2, Tue | AVL tree (1/2) |
| June 5, Fri | * lab 8 |
| June 9, Tue | AVL tree (2/2) |
| June 12, Fri | * lab 9 |
| June 16, Tue | ** final exam (TBD) |
| June 19, Fri | |

# Study Guidelines



- Read, read and read textbook
  - read regularly
  - never move on once you face a unknown word or sentence
  - use your hands to repeat examples
  - memorize definitions
  - peruse stories in boxes
  - never expect that all materials will be covered at the meeting
- Solve exercise problems by yourself
  - read the problem sentence carefully
  - write down an answer completely, and never stop at a middle
  - do have a group study
- Try best to think together (discuss) at meetings
  - participate or lose the time

# Programming Labs



- We will have 8 to 10 programming labs
  - it will be held as 3-hour class at Friday
  - it involves programming tasks
  - each student need to bring his/her own laptop

- Programming tasks
  - Two students work as a team.
  - You will have a new random partner at each lab
  - Each team is asked to complete 1 to 3 problems in 3 hours
  - Only one laptop is allowed for each team
  - Online system will be used for automated grading

# Class Policies

https://github.com/hongshin/DataStructure/wiki/Class-Policies

# Ground Rules



- Primary, students study course subjects by reading textbooks and taking the tasks given at programming labs
  - the primary purpose of a meeting is for having discussions
- Students are expected to spend at least 8 hours in a week by themselves for following up 4 hour meetings of the week
  - 8 hours excluding the time for homework and meetings
- Given programming tasks, figuring out their obligations is a crucial task to accomplish
- In team work, all team members must thoroughly understand all parts of their results
  - each member may take a part, and must study all aspects together

# Grading



- Proportion of final score attributes
  - Meeting attendance: 3%
  - Discussion (class contribution): 5% (+3%)
  - Midterm exam: 25%
    - including programming test
  - Final exam: 30%
    - including programming test
  - Programming lab: 32%
  - Homework: 5%
- Grading proportion
  (A : B : C+D+F) = (20-30% : 40-60% : 10-30%)
- Professor to the rescue
  - 14th week
  - for those who are suspected to get D or fails by 8+ absences
  - by open make-up classes or give extra homework

# Optional Assignment



- Deadline: 11:59 PM, June 27 (Sat)

- Task
  - write an essay after reading one of the recommended books (that you have not read before)
  - the essay should span over at least 5 pages
  - you should give your genuine ideas that associate the book with the topics learned in the course

- Credit: upto 3 points are added to the grading score
  - grading proportion: (A: B: C) = (20-30%: 30-50%: 30-50%)

# List of Recommended Books



- 컴퓨터 과학이 여는 세상, 이광근 (written in Korean)
- *Once Upon an Algorithm* by Martin Erwig (only English version will be counted)
- *Algorithms to Live By: The Computer Science of Human Decision* by Brian Christian et al.
- *Code: The Hidden Language of Computer Hardware and Software* by Charles Petzold
  (Code: 하드웨어와 소프트웨어에 숨어 있는 언어)
- *Automate This: How Algorithms Came To Rule Our World* by Christopher Steiner
- *Grokking Algorithms: An Illustrated Guide for Programmers and Other Curious People* by Aditya Bhargava
- *Programming Pearls*, 2/e, John Bentley (생각하는 프로그래밍)
- *From Mathematics to Genetic Programming* by Stepanov and Rose
  (알고리즘 산책: 수학에서 제네릭 프로그래밍까지)
- *Godel, Escher, Bach: An Eternal Golden Braid* by Douglas Hofstadter
  (괴델, 에셔, 바흐: 영원한 황금 노끈)
  - acceptable even if you cover only Part I (GEB)
- 튜링&괴델: 추상적 사유의 위대한 힘 (written in Korean)
- *Godel's Proof* by Ernest Nagel et al., (괴델의 증명)