CS 171: Introduction to Computer Science II

Department of Mathematics and Computer Science

Li Xiong
Today

• Meet everybody in class
• Course overview
• Course logistics
• Pre-test
Instructor and TA

• **Instructor:** Li Xiong
  – Web: [http://www.mathcs.emory.edu/~lxiong](http://www.mathcs.emory.edu/~lxiong)
  – Email: lxiong@emory.edu
  – Office Hours: TuTh 5:15-6:15pm
  – Office: MSC E412

• **TA:** Vincent Hung
  – Web: [http://www.mathcs.emory.edu/~hhung2](http://www.mathcs.emory.edu/~hhung2)
  – Email: hhung2@emory.edu
  – Office Hours: MW 10-11am
  – Office: MSC N414
About Me

• Undergraduate teaching
  – CS170 Intro to CS I
  – CS171 Intro to CS II
  – CS377 Database systems

• Graduate teaching
  – CS550 Database systems
  – CS570 Data mining
  – CS573 Data privacy and security

• Research
  – data privacy and security
  – information integration and informatics
Meet everyone in class

• Group introduction (3-5 people)
• Introducing your group
  – Names
  – Your goals for the course
  – Something interesting about your group
Today

• Meet everybody in class
• Course overview
• Course logistics
• Pre-test
What the class is about

• A continuation of CS170
• Programming and problem solving, with applications
• Algorithms and algorithm analysis – methods to solve problems
• Data structures – methods to store information
What is an algorithm

• An algorithm is a method for solving a problem expressed as a sequence of steps that is suitable for execution by a computer (machine)

• Can be expressed in
  – natural languages
  – Flowcharts
  – Pseudocode
  – programming languages
...And that, in simple terms, is how you increase your ranking on search engines.”
What is an algorithm: example

• Determine if a number n is a prime number (pseudocode and Java)

```
k = 2;
As long as k < n do
{ 1. Divide n by k
   2. If n is divisible by k, then return NO
   3. Otherwise, increase k by 1 }
return YES
```

```
int k = 2;
while ( k++ < n ) {
   if ( n%k == 0) return false;
}
return true;
```
What is a data structure

• A data structure is a way for organizing and accessing data

• Example data structures
  – Arrays
  – Trees, Graphs

• We will learn
  – Fundamental data structures and their operations
  – How to use Java’s provided data structures
  – How to implement some of them
  – How to evaluate them and decide when to use what

One-dimensional array with six elements

Tree with nine elements
Algorithms and data structures

• Algorithm + Data Structure = Program
  – An algorithm must use some data structure to store its information
  – An algorithm manipulates the data in the data structures in various ways

• To write a program
  – Design the data structures to store the information
  – Design the algorithm that uses the information to solve the problem
  – Implement the algorithm
Algorithms and data structures

“ I will, in fact, claim that the difference between a bad programmer and a good one is whether he considers his code or his data structures more important. Bad programmers worry about the code. Good programmers worry about data structures and their relationships. ”

— Linus Torvalds (creator of Linux)
Good Algorithms and Data Structures

• Good algorithms and data structures are keys to write a good program for solving a problem

• Think about maintaining a phone directory or social network
  – A large number of records
  – Add/delete/modify records
  – Missing fields in records
  – Efficient search in a giant directory
Good algorithms and data structures

• Need ways to measure “goodness” of data structures and algorithms

• Algorithm analysis
  – Running analysis, Big-O notation

• Other goodness metrics: space usage, power
Course topics

• Data structures
  – Fundamental data structures: arrays, linked lists
  – Operations (algorithms that maintain and use the data structure): search, insertion, deletion, sort
  – Abstract data types (a data structure with its associated operations): stacks, queues, trees, hash tables, graphs

• Algorithms
  – Fundamental algorithms: sort, search, recursion
  – Algorithm analysis: runtime complexity, Big-O notation

• Programming
  – Java programming techniques
  – Applications: scientific, recreational, social networks, etc.
XKCD says it better

College Activities:

- Usefulness to Career Success
  - 900 Hours of Classes
  - 400 Hours of Homework
  - One Weekend Messing with Java
Today

• Meet everybody in class
• Course overview
• Course logistics
• Pretest (does not count towards your grade)
Textbook

• Algorithms, 4th Edition, Sedgewick and Wayne
• Book site: http://algs4.cs.princeton.edu
Workload

- ~6 programming assignments (individual)
- 2 programming projects (team of up to 2 students)
- Assignment/project prep labs (not graded)
- Midterm and final exam
- Reading and quizzes
Grading

• Programming assignments  30%
• Programming projects  20%
• Midterm                20%
• Final                  25%
• Quizzes                5%
Policies

• Exams
  – All exams must be taken promptly at the required time.
  – Rescheduling midterm is possible if the request is made at least a week prior to the exam date
  – Final can not be rescheduled.

• Late assignment policy
  – Late assignment will be accepted within 3 days of the due date and penalized 10% per day. No extensions will be given.
  – 2 late assignment allowances, each can be used to turn in a single late assignment within 3 days of the due date without penalty.

• Honor code
  – College Honor Code and Departmental Policy
  – No collaboration is allowed on individual programming assignments.
  – Every program assignment must have the following comment included at the top of the file.

    /*
     * THIS CODE IS MY OWN WORK, IT WAS WRITTEN WITHOUT CONSULTING CODE WRITTEN BY OTHER STUDENTS. _Your_Name_Here_
     */
Study Strategy

• Come to class, think and participate
• Read the book or book site and play with the sample programs
• Come to office hours (TA and me)
• Start programming assignments early
• Think before program
• Enjoy and good luck!
Summary of Course Expectations

- This course will be **fun** and you will learn **a lot**, but expect to spend the **time and effort**:
  - To dig deeper into CS techniques and problem solving
  - To spend some sweat developing and debugging Java programs
- If you put in the effort, your reward will be a set of useful skills for other courses and the “real world”.
- Enjoy and good luck!
And now ...

- Meet everybody in class
- Course overview
- Course logistics
- **Pretest** (does not count towards your grade)