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: MF 11:30am – 12:30pm or by appt
  – Office: MSC E412

• **TA: Tomasz Jurczyk**
  – Email: tjurczy@emory.edu
  – Office Hours: MW 4-5pm, F 2-3pm
  – E308 (Computing lab)
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
What is an algorithm: example

• Determine if a number \( n \) is a prime number

• pseudocode:

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

• Java

\[
\text{int } k = 2;\\
\text{while ( } k++ < n \text{ ) }\\\{\\  \  \  \text{if ( } n\%k == 0) \text{ return false; }\\ \}
\text{return true;}
\]
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 implement some of them
  - How to use Java’s provided data structures
  - How to evaluate them and decide when to use what
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

- 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/projects  50%
• Midterm  20%
• Final  25%
• Quizzes  5%

8/29/2012
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!
And now ...

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