CMPS 3120

Algorithm Analysis

Dr. Chengwei Lei
CEECS
California State University, Bakersfield
The course

- Instructor: Dr. Chengwei Lei
- Office: Science III 339
- Office Hours: M/W 1:00-1:59 PM, or by appointment
- Phone: (661) 654-2102
- E-mail: clei@csub.edu
About Me

- LEI, Chengwei (雷程炜) received his Ph.D. in July 2014 from the Department of Computer Science in The University of Texas at San Antonio.

- His research interests lie in the broad area of bioinformatics, large scale data mining, network topology analysis, and clustering problems.
Short Bio

- University of Texas at San Antonio, San Antonio, Texas USA
  Ph.D., Computer Science, July 2014,
- University of Texas at San Antonio, San Antonio, Texas USA
  M.S., Computer Science, August 2008
- Beijing University of Aeronautics and Astronautics, China
  B.S., Computer Science and Technology, June 2005
- Beijing University of Aeronautics and Astronautics, China
  B.S., Applied Mathematics, June 2005
Course Description

- Analysis of the performance of algorithms;
- Discussion of programming techniques and data structures used in the writing of effective algorithms.
Textbook

- *Introduction to the Design and Analysis of Algorithms 3rd Edition,*

- Anany Levitin, Pearson, 2011.
Method of Instruction

- Lecture; Demonstration; Lab
<table>
<thead>
<tr>
<th>Component</th>
<th>Weight</th>
<th>Determination</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average of Regular Semester Exams</td>
<td>30%</td>
<td>Exams</td>
</tr>
<tr>
<td>Homework &amp; Labs</td>
<td>40%</td>
<td>As determined by instructor</td>
</tr>
<tr>
<td>Comprehensive Final Exam</td>
<td>30%</td>
<td>Comprehensive Exam</td>
</tr>
</tbody>
</table>
Late homework submissions

- Each assignment is due at the specified date at the specified time. Late submissions will be accepted at a penalty of 20% for each day the assignment is late.
Time

- MoWe  2:30 PM – 3:20 PM
  - Lecture
- Th    1:00 PM - 3:30 PM
  - Lab

Final

Friday Dec 14 2018 from 2:00 to 4:30 pm
Cheating

- You are not allowed to read, copy, or rewrite the solutions written by others *(in this or previous terms)*. Copying materials from **websites, books or any other sources** is considered equivalent to copying from another student.

- If two people are caught sharing solutions, then **both the copier and copiee** will be held **equally responsible**, which will result in zero point in homework.

- Cheating on an exam will result in failing the course.
Getting answers from the internet is CHEATING
Getting answers from your friends is CHEATING
I will send it to the Dean!
You will be nailed!
Do **NOT** get answers from others!

Discuss ideas **verbally** at a **high-level** but write up on your own.
To do well you should:

- Study with pen and paper
- Ask for help immediately
- Practice, practice, practice...
- Follow along in class rather than take notes
- Ask questions in class
- Keep up with the class
- Read the book, not just the slides
Please feel free to ask questions!

Help me know what people are not understanding

We do have a lot of material

It’s your job to slow me down
So you want to be a computer scientist?
Is your goal to be a mundane programmer?
Or a great leader and thinker?
Boss assigns task:

- Given today’s prices of pork, grain, sawdust, ...
- Given constraints on what constitutes a hotdog.
- Make the cheapest hotdog.

Everyday industry asks these questions.
Um? Tell me what to code.

With more sophisticated software engineering systems, the demand for mundane programmers will diminish.
Your answer:

- I learned this great algorithm that will work.

Soon all known algorithms will be available in libraries.

Your boss might change his mind. He now wants to make the most profitable hotdogs.
Your answer:

- I can develop a new algorithm for you.

Great thinkers will always be needed.
How do I become a great thinker?

Maybe I’ll never be...
Learn from the classical problems
Shortest path
Traveling salesman problem
Knapsack problem
There is only a handful of classical problems.
- Nice algorithms have been designed for them
- If you know how to solve a classical problem (e.g., the shortest-path problem), you can use it to do a lot of different things
  - Abstract ideas from the classical problems
  - Map your boss’ requirement to a classical problem
  - Solve with classical algorithms
  - Modify it if needed
What if you can NOT map your boss’ requirement to any existing classical problem?

How to design an algorithm by yourself?

Learn some meta algorithms

- A meta algorithm is a class of algorithms for solving similar abstract problems
- There is only a handful of them
  - E.g. divide and conquer, greedy algorithm, dynamic programming
- Learn the ideas behind the meta algorithms
  - Design a concrete algorithm for your task
What is an algorithm?

An algorithm is a sequence of unambiguous instructions for solving a problem, i.e., for obtaining a required output for any legitimate input in a finite amount of time.
What is an algorithm?

- Algorithms are the ideas behind computer programs.
- An algorithm is the thing that stays the same regardless of programming language and the computing hardware.
What is an algorithm? (cont’)

- An algorithm is a precise and unambiguous specification of a sequence of steps that can be carried out to solve a given problem or to achieve a given condition.
- An algorithm accepts some value or set of values as input and produces a value or set of values as output.
- Algorithms are closely intertwined with the nature of the data structure of the input and output values.
How to express algorithms?

Increasing precision

Nature language (e.g. English)

Pseudocode

Real programming languages

Ease of expression

Describe the *ideas* of an algorithm in nature language.
Use pseudocode to clarify sufficiently tricky details of the algorithm.
How to express algorithms?

To understand / describe an algorithm:
Get the **big idea** first.
Use pseudocode to clarify sufficiently **tricky details**

- Increasing precision
  - Nature language (e.g. English)
  - Pseudocode
  - Real programming languages

- Ease of expression
What is a program?

- How to cook?
- The algorithm
- Are you a programmer?
This is the pseudocode for a game of Monopoly

Main Procedure Monopoly_Game
Hand out each player's initial money.
Decide which player goes first.
Repeat
   Call Procedure Monopoly_Move for next player.
   Decide if this player must drop out.
Until all players except one have dropped out.
Declare the surviving player to be the winner.

Procedure Monopoly_Move
Begin one's move.
Throw the dice.
Move the number of spaces on the board shown on the dice.
If the token landed on "Go to Jail,"
   then go there immediately.
Else if the token landed on "Chance" or "Community Chest,"
   then draw a card and follow its instructions.
Else
   follow the usual rules for the square (buying property,
   paying rent, collecting $200 for passing "Go", etc.).
End one's move.
Flowcharts

<table>
<thead>
<tr>
<th>Name</th>
<th>Symbol</th>
<th>Use in flowchart</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oval</td>
<td><img src="image" alt="Oval" /></td>
<td>Denotes the beginning or end of a program.</td>
</tr>
<tr>
<td>Flow line</td>
<td><img src="image" alt="Flow line" /></td>
<td>Denotes the direction of logic flow in a program.</td>
</tr>
<tr>
<td>Parallelogram</td>
<td><img src="image" alt="Parallelogram" /></td>
<td>Denotes either an input operation (e.g., INPUT) or an output operation (e.g., PRINT).</td>
</tr>
<tr>
<td>Rectangle</td>
<td><img src="image" alt="Rectangle" /></td>
<td>Denotes a process to be carried out (e.g., an addition).</td>
</tr>
<tr>
<td>Diamond</td>
<td><img src="image" alt="Diamond" /></td>
<td>Denotes a decision (or branch) to be made. The program should continue along one of two routes (e.g., IF/THEN/ELSE).</td>
</tr>
</tbody>
</table>

Start → Throw the dice → Move token number of spaces equal to sum of dice

Move token immediately to Jail square

Did token land on Go to Jail square?

NO

Follow instructions on card drawn from proper deck

Did token land on Chance or Community Chest square?

NO

Let next player begin her turn

Follow usual rules for square where token landed: buying property, paying rent, etc.
"Hey Jude" is a song by the English rock band the Beatles, written by Paul McCartney and credited to Lennon–McCartney.
hey Jude

make it bad
be afraid
let me down

remember to
let her into your heart
let her under your skin

then you
can start
begin
to make it better

better better better better better waaaaa

na
Flowcharts details
Problem: Find $\text{gcd}(m,n)$, the greatest common divisor of two nonnegative, not both zero integers $m$ and $n$