CMPS 3120

Algorithm Analysis

Dr. Chengwei Lei
CEECS
California State University, Bakersfield
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.
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

Start

end
Traveling salesman problem
Knapsack problem

$4 12 \text{ kg}
$2 2 \text{ kg}
$2 1 \text{ kg}
$1 1 \text{ kg}
$10 4 \text{ kg}

15 \text{ kg}
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?

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></td>
<td>Denotes the beginning or end of a program.</td>
</tr>
<tr>
<td>Flow line</td>
<td></td>
<td>Denotes the direction of logic flow in a program.</td>
</tr>
<tr>
<td>Parallelogram</td>
<td></td>
<td>Denotes either an input operation (e.g., INPUT) or an output operation (e.g., PRINT).</td>
</tr>
<tr>
<td>Rectangle</td>
<td></td>
<td>Denotes a process to be carried out (e.g., an addition).</td>
</tr>
<tr>
<td>Diamond</td>
<td></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>
Hey Jude

- "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
- take a sad song and make it better
- don’t
- be afraid
- you were made to go out and get her
- let me down
- you have found her, now go and get her

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

then you
- can start
- to make it better
- begin

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 $n$