CSC 225

Algorithms and Data Structures I
Fall 2014
Rich Little

Lectures and Labs

Rich Little
- E-mail: rlittle@uvic.ca
- Voice: 472-5752
- Office: ECS 516
- Office hours:
MR 10:00 pm – 11:30 pm

- Lectures
  - A01 TWF 10:30 – 11:20 am BWC A104
- Labs
  - Fang Dong
  - Labs start week of Sep 8, 2014
  - B01 Wednesday 12:30 – 1:20 pm ECS 258
  - B02 Wednesday 1:30 – 2:20 pm ECS 258
  - B03 Wednesday 2:30 – 3:20 pm ECS 258
  - B04 Thursday 2:30 – 3:20 pm ECS 258
  - B05 Thursday 3:30 – 4:20 pm ECS 258
  - Check UVic website

- Course Web pages
  - Official Webpage on the Department Website
  - Detailed Course Website on ConneX

Administrative Officer Announcements

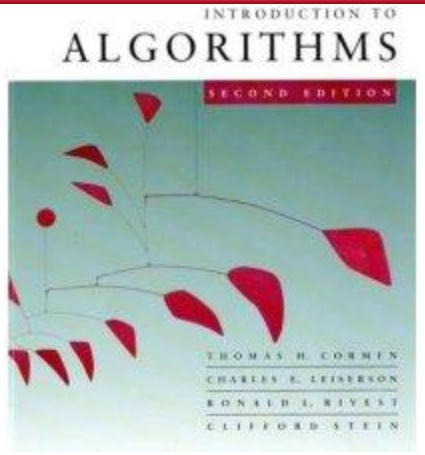
- CSC Undergraduate Officer is Sue Butler
  E-mail: cscadvisor@uvic.ca Office: ECS 512
- Any student who has registered in CSC 225 and **does not** have the required pre-requisites and no waiver **must drop the class**. Otherwise: student will be dropped and a pre-requisite drop is recorded on the student's record.
- Taking the course more than twice:

  you must request, in writing, permission from the Chair of the Department and the Dean of the Faculty to be allowed to stay registered in the class (University Rule). The letter should be given to Sue Butler, Undergraduate Advisor. Otherwise: student will be dropped from class.
- Always use and check your UVic e-mail account and use CSC 225 as part of the subject line.
- Do not send messages from other accounts (such messages are filtered and discarded).
- Register for labs!

Books

- **Required Textbook**
  R. Sedgewick and K. Wayne
  Algorithms, *Fourth Edition*
  Addison-Wesley, Toronto, 2011
  ISBN: 0-321-57351-X
- http://algs4.cs.princeton.edu/home/

- **Optional Textbook (online)**
  T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein. *Introduction to Algorithms.*
  MIT Press (2001), 2nd edition.



Lectures and Labs

- **Attendance of Lectures**
  - Essential for doing well on assignments and exams
- **Labs**
  - Extra details and hints on assignments

Evaluation

Assignments & Quizzes	40%
Midterm	15%
Final	45%

- Marks will be posted on the web by student id
  - If you do not want your marks to be posted in this manner, notify the instructor by e-mail (rlittle@uvic.ca) before Sep 8, 2014
- Midterm exam will be in-class, one hour, closed books, closed notes, no calculators, no gadgets **Wednesday, October 15, 2014**
- The final exam will be three hours, closed books, closed notes, no calculators, no gadgets scheduled by the registrar

Assignment Schedule

A1	Sep 26, 2014 (due date)
A2	Oct 10, 2014
A3	Oct 24, 2014
A4	Nov 7, 2014
A5	Nov 28, 2014

Assignments

- **Late submissions are <u>not</u> accepted**
  - ➢ if valid excuse (e.g., doctor's statement), raise weight of other assignments to compensate
- **Programming: work in the labs or at home**
  - ➢ use your favorite Java environment
  - ➢ Textbook's booksite has supplemental classes and data
    - http://algs4.cs.princeton.edu/home/
- **Cheating: zero-tolerance policy**
  - ➢ first time fail assignment, second time fail course

Reading Assignment

- **Chapter 1 – 1.1, 1.2 in Sedgewick and Wayne**
- Algorithm wiki
  - http://en.wikipedia.org/wiki/Algorithm
  - History: Development of the notion of "algorithm"
- Data structures wiki
  - http://en.wikipedia.org/wiki/Data_structure
  - http://en.wikipedia.org/wiki/List_of_data_structures
  - http://en.wikibooks.org/wiki/Data_Structures

Prerequisites

- **CSC 115**
  - Basic Java knowledge and programming skills
  - Object-oriented programming
  - Basics in fundamental algorithms and data structures as discussed in CSC 115

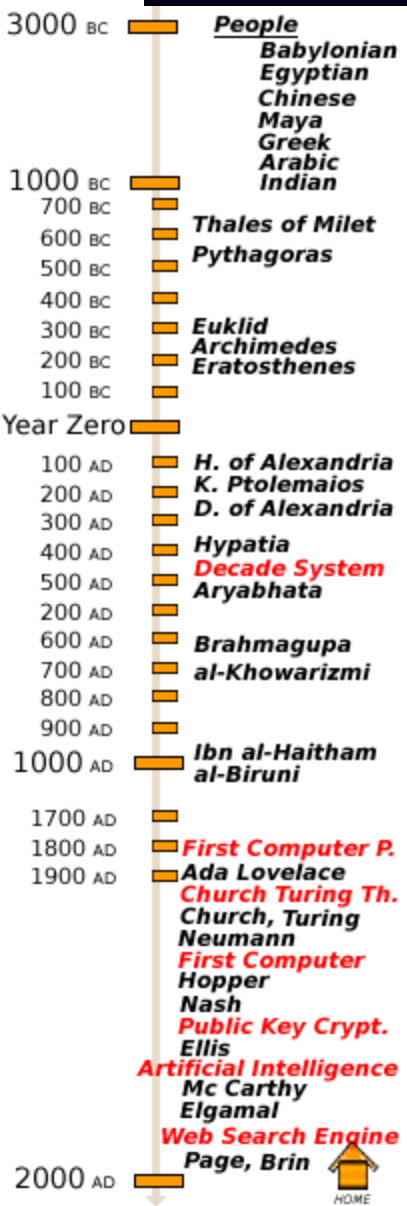
Lecture Notes

- **Acknowledgments**
  - Most of the slides for this course were prepared by Dr. Ulrike Stege and Dr. Kevin Wayne. Thank you!!
- Consider posted lecture slides as *additional* information
- **Note**
  - Not all materials required for the midterm and final exams are on the lecture slides

Questions?

- Regarding questions on lectures, assignments, algorithms, data structures, programming, Java, etc. consult in the following order:
  - Study group, book, book website
  - ConneX web page
  - Lab instructor
  - Instructor

History of Algorithms



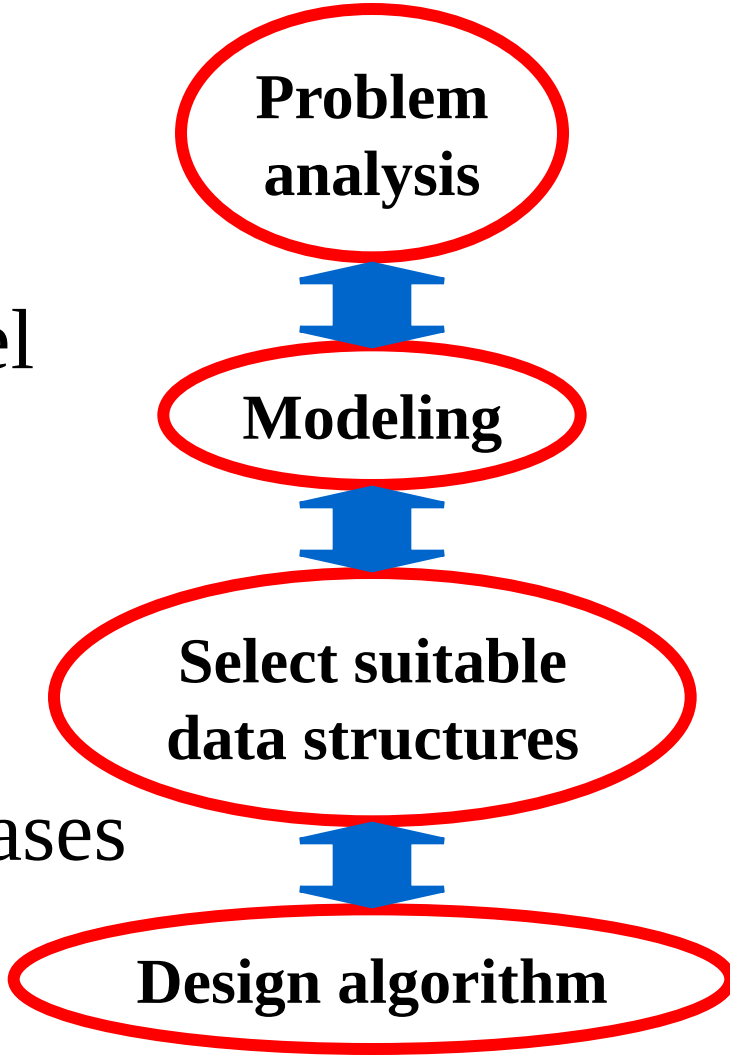
- The word algorithm can be traced back to the 9th century to the Persian scientist, astronomer and mathematician Abdullah Muhammad bin Musa al-Khwarizmi, often cited as "The father of Algebra"
- In the 12th century one of his books was translated into Latin, where his name was rendered as "Algorithmi"
- Algorithms are everywhere. They have been developed to ease our daily life from calculating algorithms, to artificial intelligence and molecular biology. The searching and sorting algorithms embodied in Google are a good example of our daily use of algorithms.
- In the age of information, people are inundated with data. With the aid of powerful algorithms and data structures, we can make sense of volumes of data that come in many forms: text, numbers, images, video, audio.

- **History of Algorithms**
  http://cs-exhibitions.uni-klu.ac.at/index.php?id=193

Algorithmics

- Problem analysis
- Design an appropriate model
- Select data structures
- Select algorithms

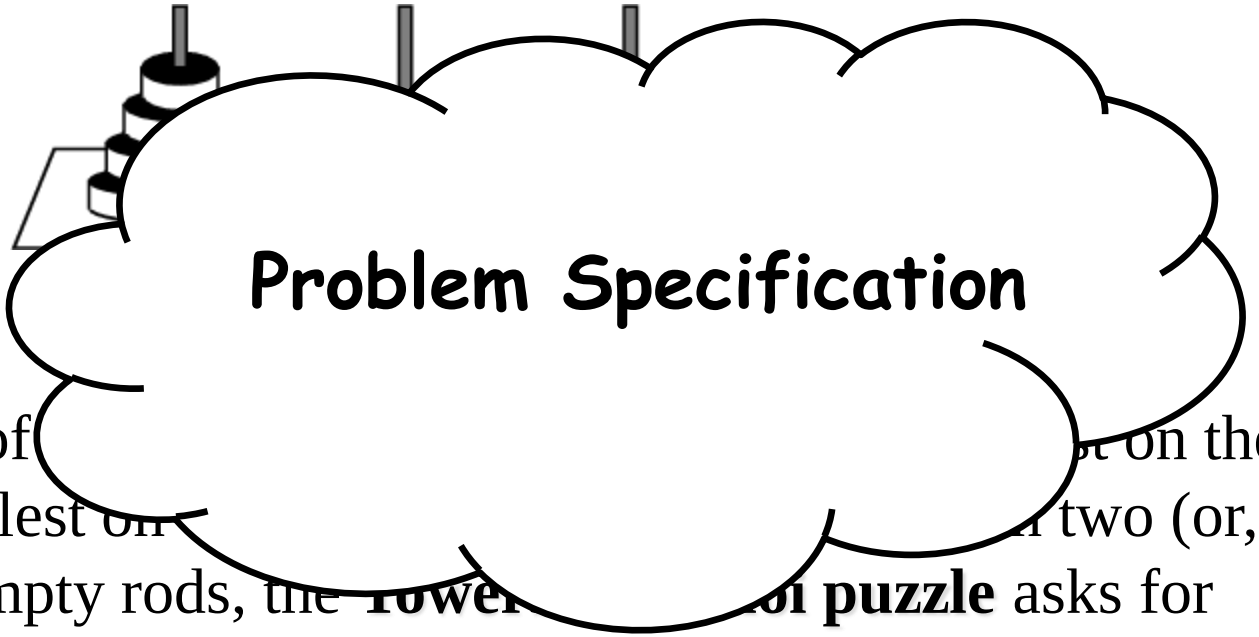
- Iterate over these design phases



Algorithm Design Techniques

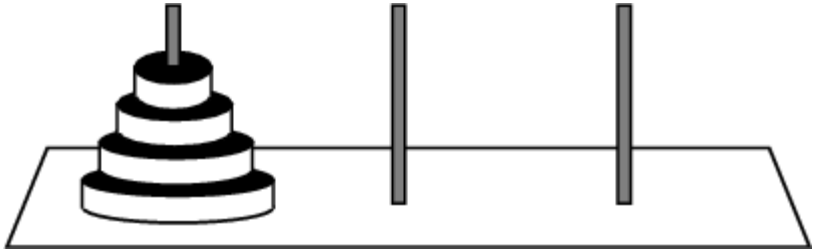
- **Algorithm Design Techniques**
  - Greedy algorithms
    - Local optimums lead to global optimum
  - Divide and conquer
    - Recursively subdivide problem
  - Backtracking
    - Technique for finding all solutions
  - Dynamic programming
    - Incrementally build complete solution

Towers of Hanoi



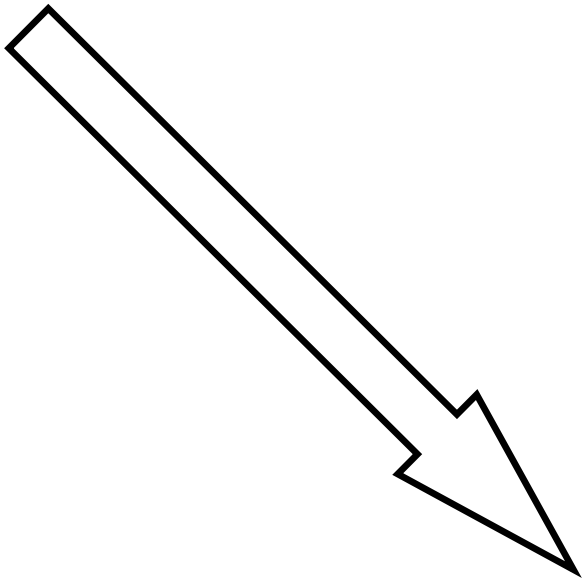
Given a stack of [illegible] on the bottom to smallest on [illegible] two (or, in general *r*) empty rods, the **Tower[illegible]oi puzzle** asks for the *minimum number* of disk-moves required to move the stack from one rod to another. **[illegible]e moves are not allowed only if they place larger disks on top of smaller disks.**

Which Type of Algorithm Solves the Towers of Hanoi Problem?



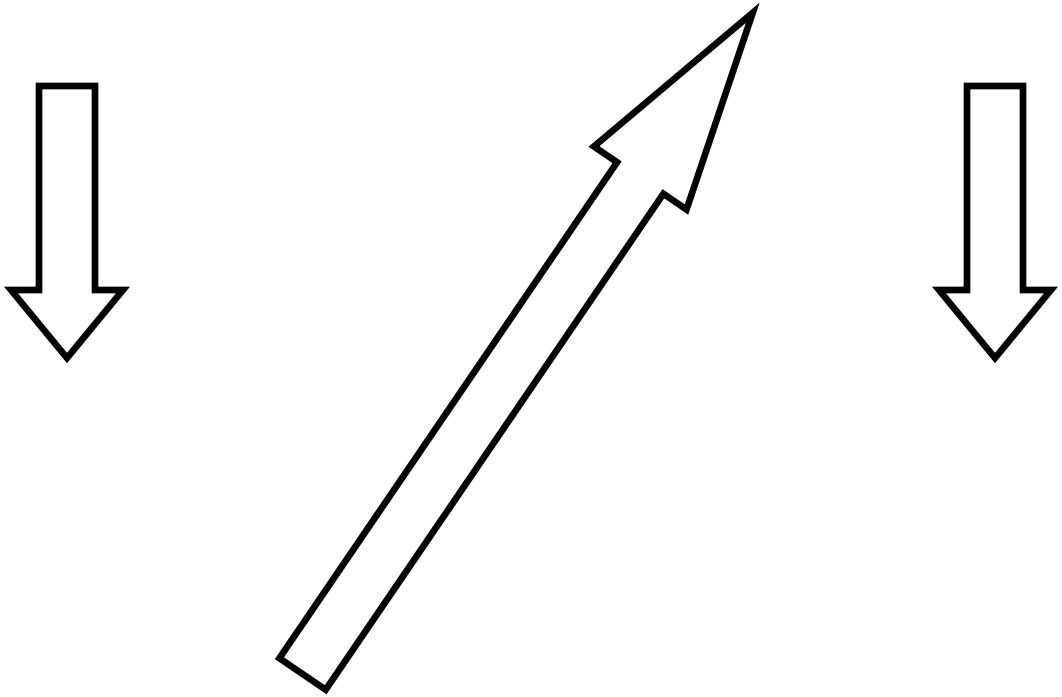
Recursion and Backtracking

Towers of Hanoi



$n = 4$

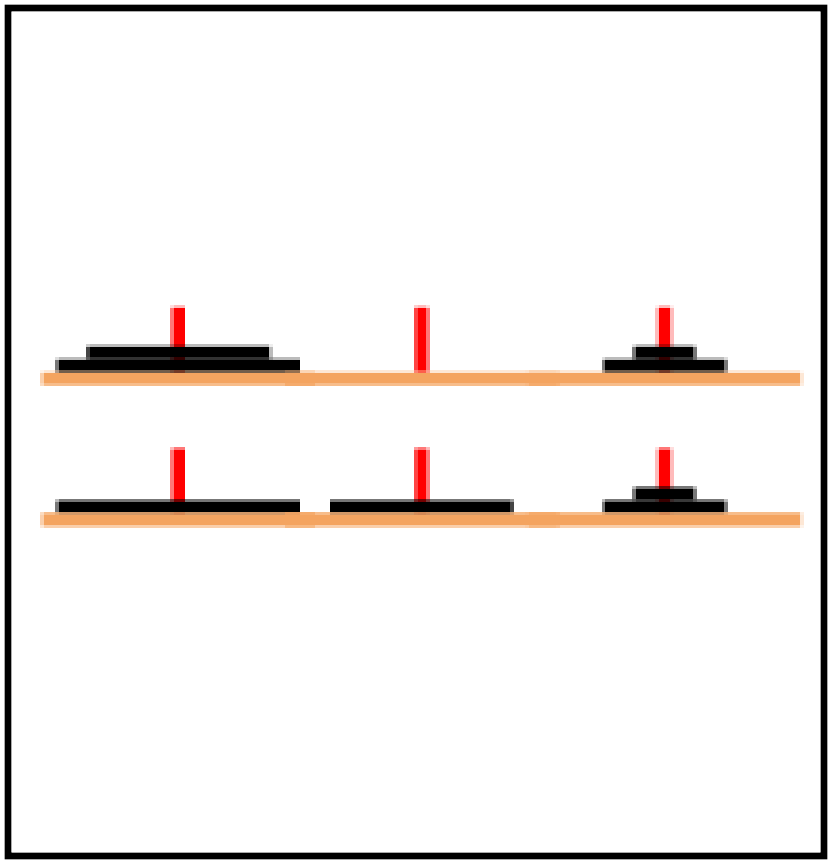
Solving for $n = 3$ helps!

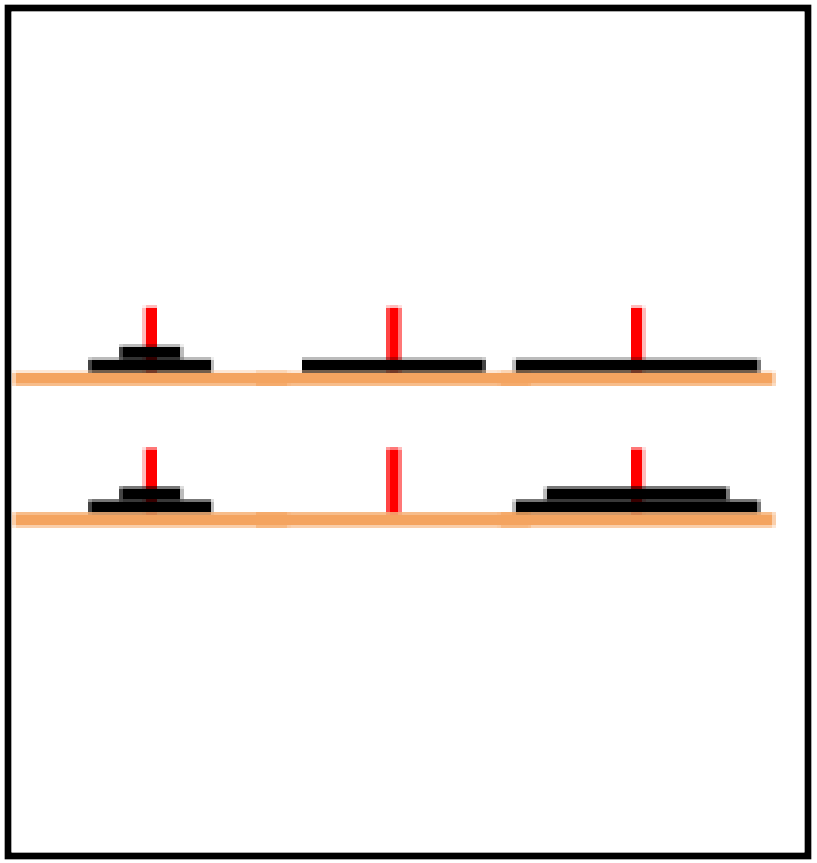


$n = 3$

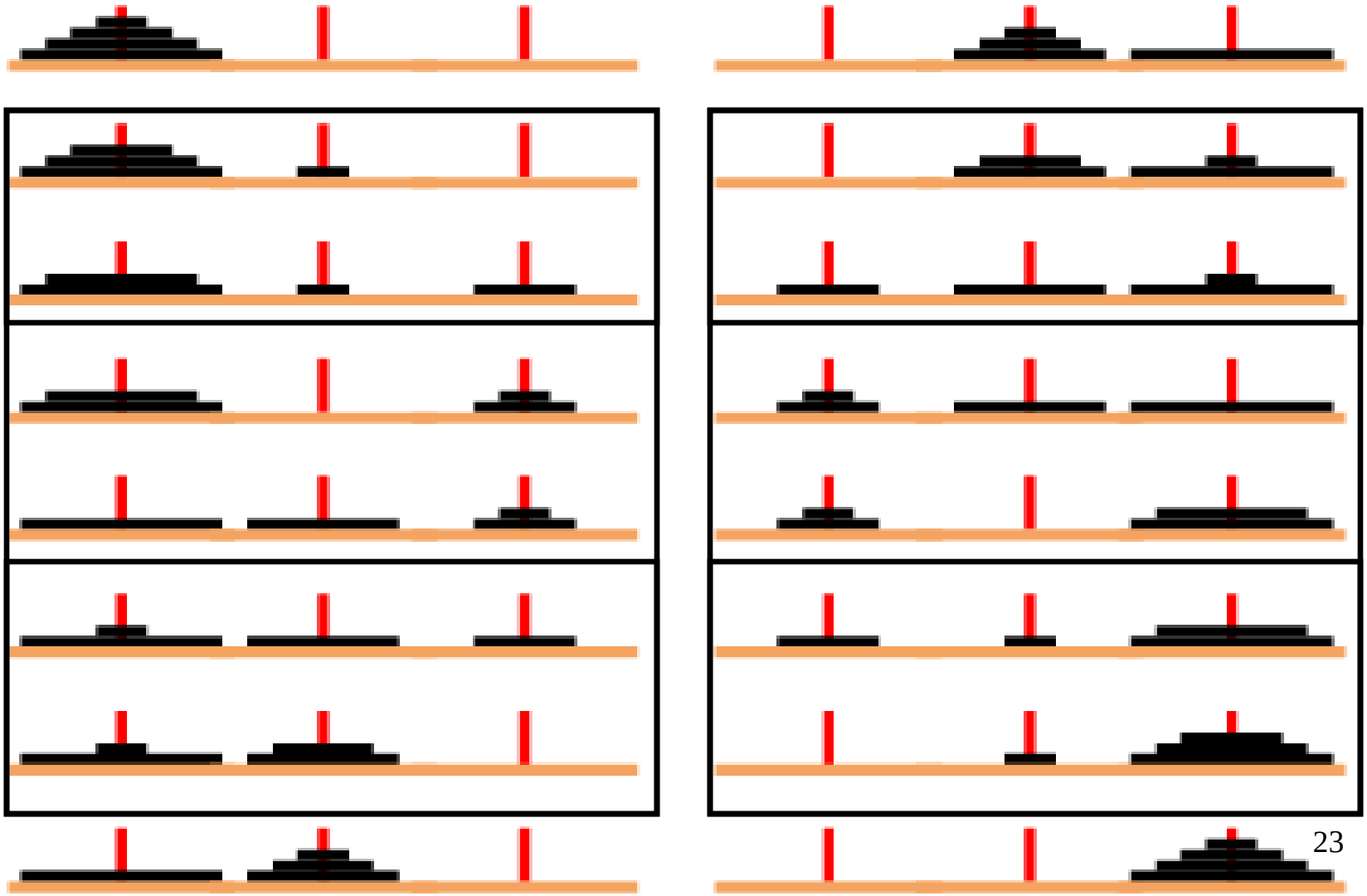
- Solving for $n = 2$ helps!

$n = 2$

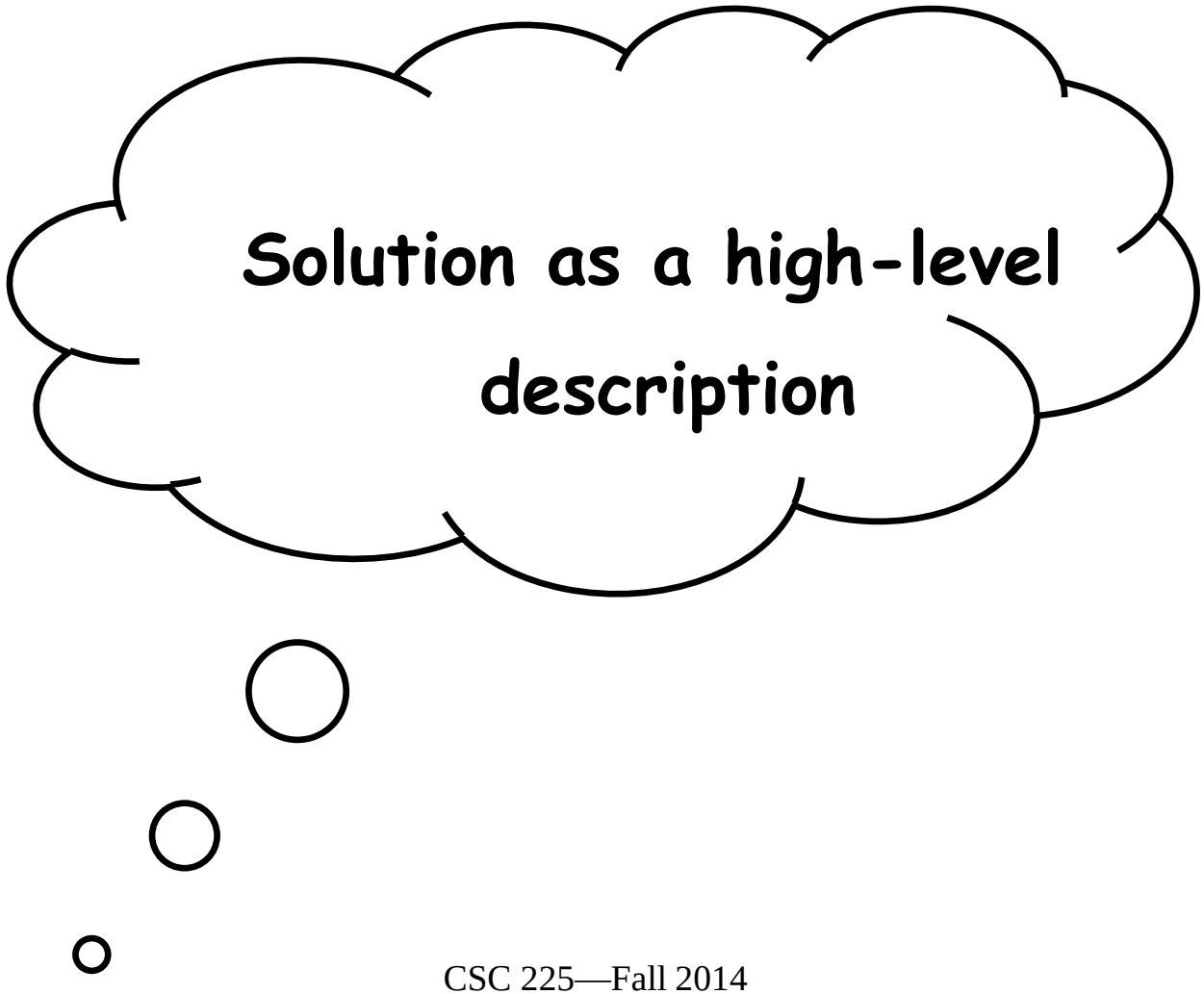




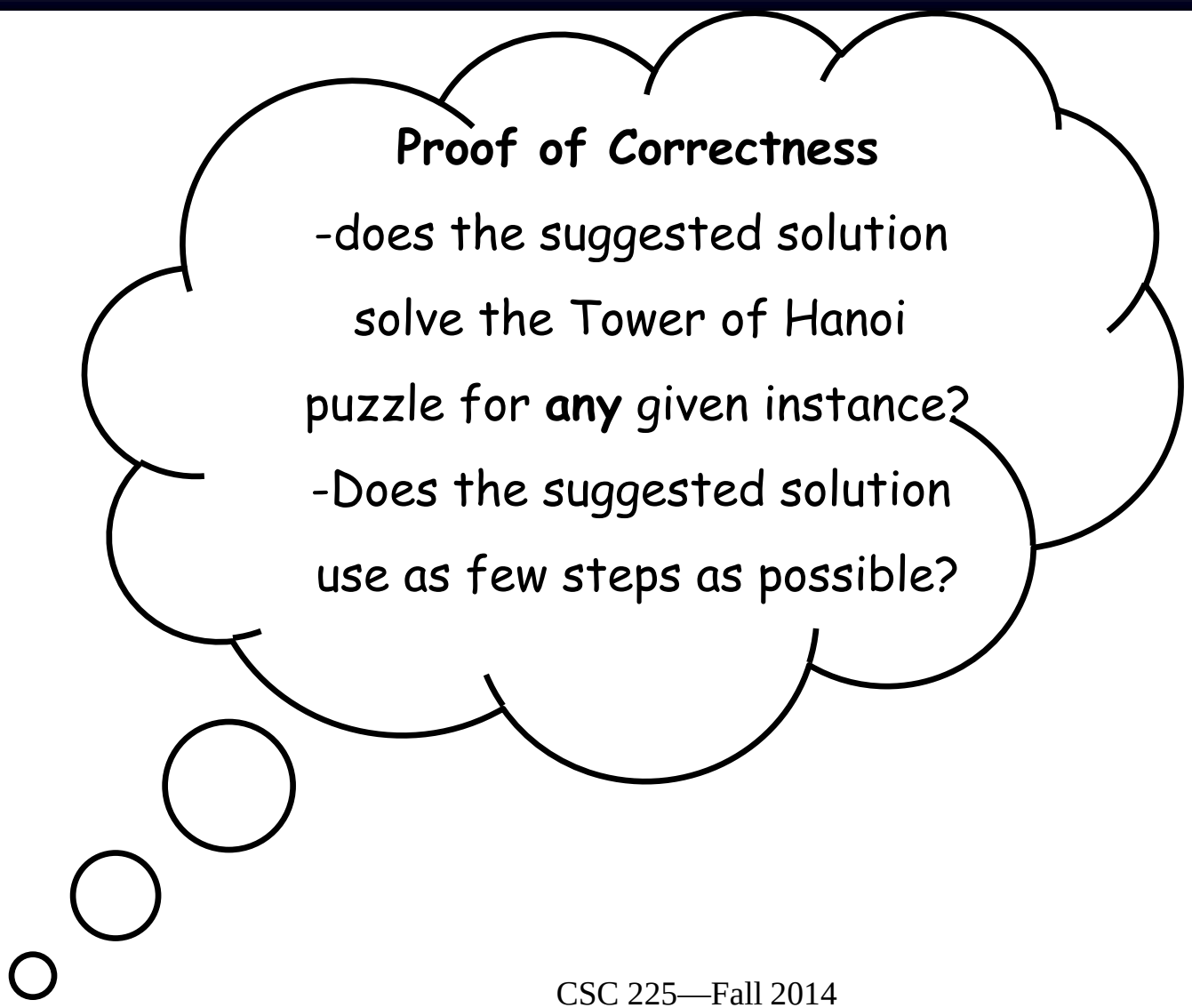
Solution for $n = 4$ (15 moves required)



Still missing



Still missing



Questions

- How many moves are required for $n = 5$?
- How many moves are required for n discs?
- Does the number of moves change if 4 rods (instead of 3 rods) are allowed?
  - $n = 1$
  - $n = 2$
  - $n = 3$
  - $n = 4$
  - $n = 5$