Algorithmic Problem Solving

Mohan Paturi

University of California, San Diego

SPIS 2014

Today's Plan

1. World of algorithms
2. Divide-and-conquer method
3. Tromino puzzle
4. Counterfeit among 8 coins
5. Course logistics
6. Homework

What is APS?

- Introduction to basic algorithmic strategies for solving problems.

What is APS?

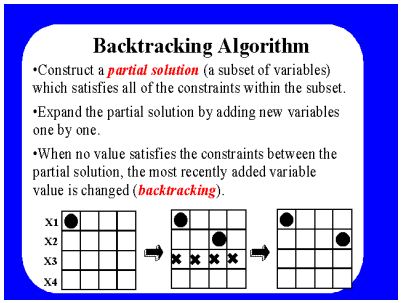
- Introduction to basic algorithmic strategies for solving problems.
- Emphasis on writing solutions precisely and coherently.

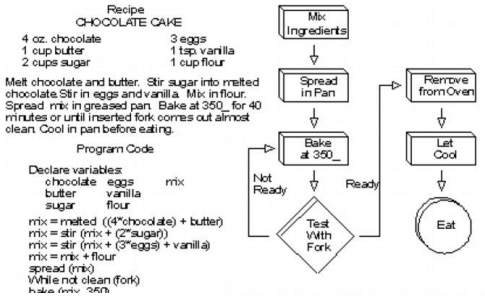
What is APS?

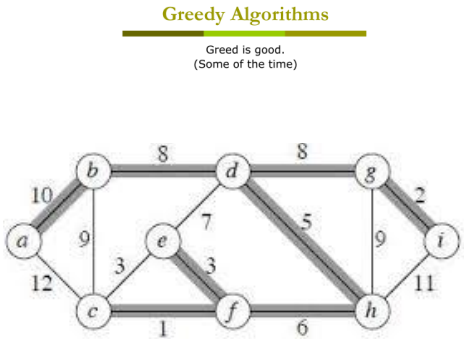
- Introduction to basic algorithmic strategies for solving problems.
- Emphasis on writing solutions precisely and coherently.
- Practice discovering algorithms and describing them.

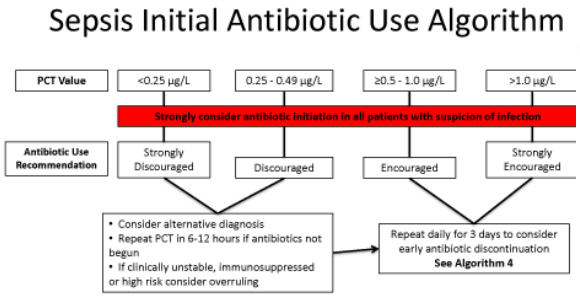
What is APS?

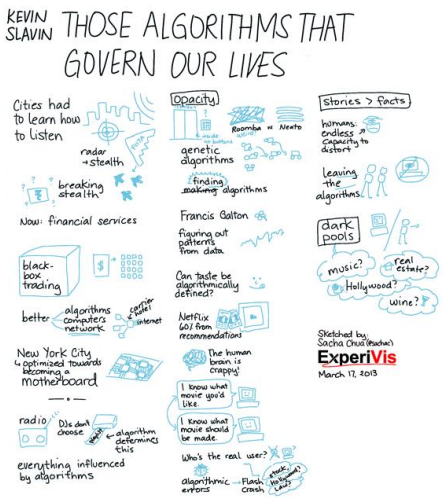
- Introduction to basic algorithmic strategies for solving problems.
- Emphasis on writing solutions precisely and coherently.
- Practice discovering algorithms and describing them.
- Analyze algorithms











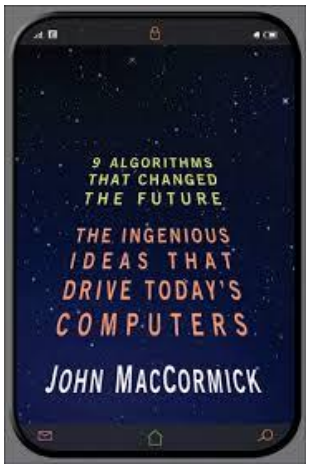


Figure: World of Algorithms

Divide-and-conquer method

- Divide a problem into several subproblems (of constant fraction size).

Divide-and-conquer method

- Divide a problem into several subproblems (of constant fraction size).
- Solve each subproblem recursively.

Divide-and-conquer method

- Divide a problem into several subproblems (of constant fraction size).
- Solve each subproblem recursively.
- Combine the solutions to the subproblems to arrive at the solution to the problem.

Divide-and-conquer method

- Divide a problem into several subproblems (of constant fraction size).
- Solve each subproblem recursively.
- Combine the solutions to the subproblems to arrive at the solution to the problem.
- Balance the sizes of the subproblems as much as possible.

Tromino Puzzle

- Cover a $2^n \times 2^n$ ($n \geq 1$) board missing one square with right tromnioes, which are L-shaped tiles formed by three adjacent squares.

A Protocol for Solving Problems

- Analyze the problem

Tromino Puzzle: Problem Analysis

- Cover a $2^n \times 2^n$ $(n \geq 1)$ board missing one square with right tromnioes, which are L-shaped tiles formed by three adjacent squares.

Tromino Puzzle: Problem Analysis

- Cover a $2^n \times 2^n$ $(n \geq 1)$ board missing one square with right tromnioes, which are L-shaped tiles formed by three adjacent squares.
- The missing square can be any of the board squares.

Tromino Puzzle: Problem Analysis

- Cover a $2^n \times 2^n$ $(n \geq 1)$ board missing one square with right tromnioes, which are L-shaped tiles formed by three adjacent squares.
- The missing square can be any of the board squares.
- Trominoes should cover all the squares except the missing ones with no overlaps.

Tromino Puzzle: Problem Analysis

- Cover a $2^n \times 2^n$ $(n \geq 1)$ board missing one square with right tromnioes, which are L-shaped tiles formed by three adjacent squares.
- The missing square can be any of the board squares.
- Trominoes should cover all the squares except the missing ones with no overlaps.
- A right tromino can also be viewed as a 2×2 board with exactly one missing square.

A Protocol for Solving Problems

- Analyze the problem
- Work out concrete examples; make note of boundary cases

A Protocol for Solving Problems

- Analyze the problem
- Work out concrete examples; make note of boundary cases
- Brainstorm about the problem

A Protocol for Solving Problems

- Analyze the problem
- Work out concrete examples; make note of boundary cases
- Brainstorm about the problem
- Design the algorithm

A Protocol for Solving Problems

- Analyze the problem
- Work out concrete examples; make note of boundary cases
- Brainstorm about the problem
- Design the algorithm
- Analyze the algorithm

A Protocol for Solving Problems

- Analyze the problem
- Work out concrete examples; make note of boundary cases
- Brainstorm about the problem
- Design the algorithm
- Analyze the algorithm
- Write the solution

A Protocol for Solving Problems

- Analyze the problem
- Work out concrete examples; make note of boundary cases
- Brainstorm about the problem
- Design the algorithm
- Analyze the algorithm
- Write the solution
- Revise

A Protocol for Solving Problems

- Analyze the problem
- Work out concrete examples; make note of boundary cases
- Brainstorm about the problem
- Design the algorithm
- Analyze the algorithm
- Write the solution
- Revise
- Reflect

A Protocol for Solving Problems

- Analyze the problem
- Work out concrete examples; make note of boundary cases
- Brainstorm about the problem
- Design the algorithm
- Analyze the algorithm
- Write the solution
- Revise
- Reflect
- Generalize

A Protocol for Solving Problems

- Analyze the problem
- Work out concrete examples; make note of boundary cases
- Brainstorm about the problem
- Design the algorithm
- Analyze the algorithm
- Write the solution
- Revise
- Reflect
- Generalize
- Learn about the history and applications

Divide-and-conquer Algorithm for the Tromino Puzzle

- If $n = 1$, we have a 2×2 square with a missing square, which can be covered with one tromino.

Divide-and-conquer Algorithm for the Tromino Puzzle

- If $n = 1$, we have a 2×2 square with a missing square, which can be covered with one tromino.
- If $n \geq 2$, we divide the board into four smaller boards, each of size $2^{n-1} \times 2^{n-1}$.

Divide-and-conquer Algorithm for the Tromino Puzzle

- If $n = 1$, we have a 2×2 square with a missing square, which can be covered with one tromino.
- If $n \geq 2$, we divide the board into four smaller boards, each of size $2^{n-1} \times 2^{n-1}$.
- However, of the four smaller boards, three of them do not miss any squares.

Divide-and-conquer Algorithm for the Tromino Puzzle

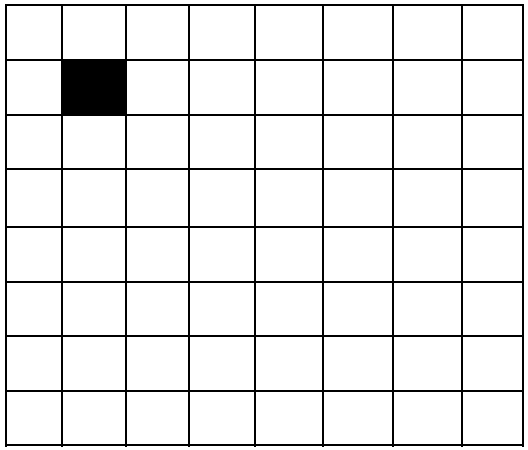
- If $n = 1$, we have a 2×2 square with a missing square, which can be covered with one tromino.
- If $n \geq 2$, we divide the board into four smaller boards, each of size $2^{n-1} \times 2^{n-1}$.
- However, of the four smaller boards, three of them do not miss any squares.
- We use an appropriately oriented tromino to cover the three corner squares (which meet at the center of the board) of the three smaller boards.

Divide-and-conquer Algorithm for the Tromino Puzzle

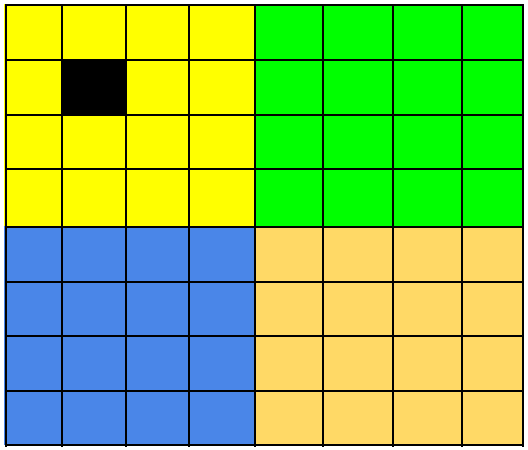
- If $n = 1$, we have a 2×2 square with a missing square, which can be covered with one tromino.
- If $n \geq 2$, we divide the board into four smaller boards, each of size $2^{n-1} \times 2^{n-1}$.
- However, of the four smaller boards, three of them do not miss any squares.
- We use an appropriately oriented tromino to cover the three corner squares (which meet at the center of the board) of the three smaller boards.
- We now have four smaller boards, each with exactly one missing square.

Divide-and-conquer Algorithm for the Tromino Puzzle

- If $n = 1$, we have a 2×2 square with a missing square, which can be covered with one tromino.
- If $n \geq 2$, we divide the board into four smaller boards, each of size $2^{n-1} \times 2^{n-1}$.
- However, of the four smaller boards, three of them do not miss any squares.
- We use an appropriately oriented tromino to cover the three corner squares (which meet at the center of the board) of the three smaller boards.
- We now have four smaller boards, each with exactly one missing square.
- We cover them recursively with L-shaped trominoes.



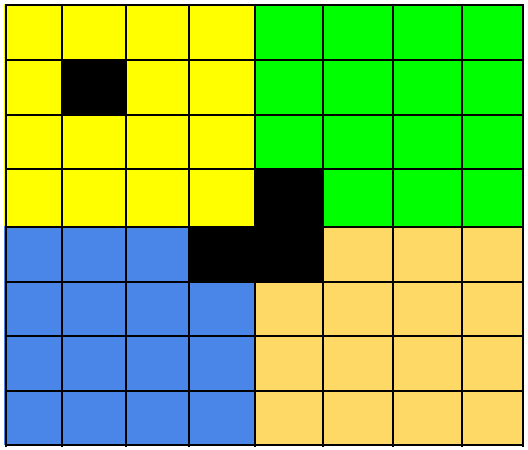
8 x 8 Board with one missing square



8 x 8 Board with one missing square

Division into 4 equal quadrants (4 x 4 boards)

The subproblems of covering the quadrants are not exactly the same type as the original problem

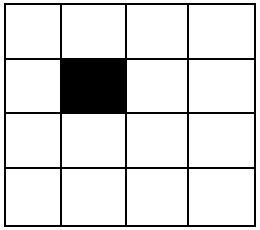


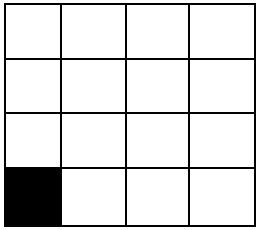
8 x 8 Board with one missing square

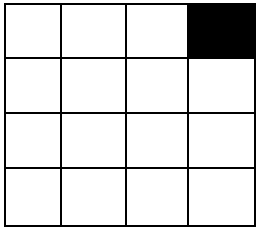
Division into 4 equal smaller size boards (4 x 4 boards)

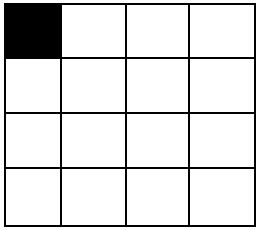
The subproblems of covering the smaller boards are not exactly the same type as the original problem since three of them do not have a missing square

Use a tromino to cover the three corner squares of the three smaller boards which do not have a missing square. The three corner squares meet at the center of the board.









4 subproblems: 4 4x4 boards each with one missing square

Goals

- Apply basic algorithmic paradigms to solve problems

Goals

- Apply basic algorithmic paradigms to solve problems
- Write algorithms with precision, logical coherence, and rigor

Goals

- Apply basic algorithmic paradigms to solve problems
- Write algorithms with precision, logical coherence, and rigor
- Analyze easy algorithm, and

Goals

- Apply basic algorithmic paradigms to solve problems
- Write algorithms with precision, logical coherence, and rigor
- Analyze easy algorithm, and
- Parse rigorously written material

Instructors and Mentors

- Instructors: Mohan Paturi, Hao Ye, Mark Land, Giorgio Quer

Instructors and Mentors

- Instructors: Mohan Paturi, Hao Ye, Mark Land, Giorgio Quer
- Tutors: Russell Impagliazzo, Nathan Ng, Marco Carmosino, Daniel Moeller

Topics

- Divide-and-conquer and searching (Mohan Paturi)
  Lecture: August 20th, 2014; Homework due Saturday August 23rd, 11:59 PM
  Workshop: Monday, August 25th

Topics

- Divide-and-conquer and searching (Mohan Paturi)
  Lecture: August 20th, 2014; Homework due Saturday August 23rd, 11:59 PM
  Workshop: Monday, August 25th
- Transform and conquer (Hao Ye) Lecture: August 27th, 2014; Homework due Saturday August 30th, 11:59 PM
  Workshop: Tuesday, September 2, 3-4 PM (Section 1) and 4-5 PM (Section 2), CSE 4140

Topics

- Divide-and-conquer and searching (Mohan Paturi)
  Lecture: August 20th, 2014; Homework due Saturday August 23rd, 11:59 PM
  Workshop: Monday, August 25th
- Transform and conquer (Hao Ye) Lecture: August 27th, 2014; Homework due Saturday August 30th, 11:59 PM
  Workshop: Tuesday, September 2, 3-4 PM (Section 1) and 4-5 PM (Section 2), CSE 4140
- Greedy method and iterative improvement (Mark Land)
  Lecture: September 3rd, 2014; Homework due Saturday August 6th, 11:59 PM
  Workshop: Monday, September 8th

Topics

- Divide-and-conquer and searching (Mohan Paturi)
  Lecture: August 20th, 2014; Homework due Saturday August 23rd, 11:59 PM
  Workshop: Monday, August 25th
- Transform and conquer (Hao Ye) Lecture: August 27th, 2014; Homework due Saturday August 30th, 11:59 PM
  Workshop: Tuesday, September 2, 3-4 PM (Section 1) and 4-5 PM (Section 2), CSE 4140
- Greedy method and iterative improvement (Mark Land)
  Lecture: September 3rd, 2014; Homework due Saturday August 6th, 11:59 PM
  Workshop: Monday, September 8th
- Backtracking and dynamic programming (Giorgio Quer)
  Lecture: September 10th, 2014; Homework due Saturday August 13th, 11:59 PM
  Workshop: Monday, September 15th

Topics

- Final lecture: Each group will make a 15-minute presentation about their model.

About the Workshops

- Tutorials are every Monday morning (except for Labor Day)

About the Workshops

- Tutorials are every Monday morning (except for Labor Day)
- There will be a short summary lecture

About the Workshops

- Tutorials are every Monday morning (except for Labor Day)
- There will be a short summary lecture
- There will be one mentor for every group.

About the Workshops

- Tutorials are every Monday morning (except for Labor Day)
- There will be a short summary lecture
- There will be one mentor for every group.
- Each mentor will spend an hour with the group going over the homework and discussing advanced problems

About the Workshops

- Tutorials are every Monday morning (except for Labor Day)
- There will be a short summary lecture
- There will be one mentor for every group.
- Each mentor will spend an hour with the group going over the homework and discussing advanced problems
- Workshops are informal.

Expectations

- Think about the ideas, problems, and solutions discussed in the class and recreate them for yourself

Expectations

- Think about the ideas, problems, and solutions discussed in the class and recreate them for yourself
- Critically analyze the posted solutions.

Expectations

- Think about the ideas, problems, and solutions discussed in the class and recreate them for yourself
- Critically analyze the posted solutions.
- Use the posted solutions as a guide to your writing.

Expectations

- Think about the ideas, problems, and solutions discussed in the class and recreate them for yourself
- Critically analyze the posted solutions.
- Use the posted solutions as a guide to your writing.
- Complete the homework and turn in the solutions using turnin. Homework is due on Saturdays at midnight

Expectations

- Think about the ideas, problems, and solutions discussed in the class and recreate them for yourself
- Critically analyze the posted solutions.
- Use the posted solutions as a guide to your writing.
- Complete the homework and turn in the solutions using turnin. Homework is due on Saturdays at midnight
- Final homework submissions will be on Wednesdays at the beginning of the class. Print (or write) your homework and bring it to the class on Wednesdays.

Expectations

- Think about the ideas, problems, and solutions discussed in the class and recreate them for yourself
- Critically analyze the posted solutions.
- Use the posted solutions as a guide to your writing.
- Complete the homework and turn in the solutions using turnin. Homework is due on Saturdays at midnight
- Final homework submissions will be on Wednesdays at the beginning of the class. Print (or write) your homework and bring it to the class on Wednesdays.
- Advanced and challenge problems are entirely optional.

Expectations

- Think about the ideas, problems, and solutions discussed in the class and recreate them for yourself
- Critically analyze the posted solutions.
- Use the posted solutions as a guide to your writing.
- Complete the homework and turn in the solutions using turnin. Homework is due on Saturdays at midnight
- Final homework submissions will be on Wednesdays at the beginning of the class. Print (or write) your homework and bring it to the class on Wednesdays.
- Advanced and challenge problems are entirely optional.
- You are encouraged to brainstorm with your colleagues about the problems

Expectations

- Think about the ideas, problems, and solutions discussed in the class and recreate them for yourself
- Critically analyze the posted solutions.
- Use the posted solutions as a guide to your writing.
- Complete the homework and turn in the solutions using turnin. Homework is due on Saturdays at midnight
- Final homework submissions will be on Wednesdays at the beginning of the class. Print (or write) your homework and bring it to the class on Wednesdays.
- Advanced and challenge problems are entirely optional.
- You are encouraged to brainstorm with your colleagues about the problems
- However, write your solutions entirely by yourself without any help from others.

Expectations

- You are not expected to consult any other resource, except for the resources that are linked from this page. If you believe that resources are helpful to you and others, consult the instructor.

Expectations

- You are not expected to consult any other resource, except for the resources that are linked from this page. If you believe that resources are helpful to you and others, consult the instructor.
- Please read and revise your homework at least three times. Note that better writing always requires multiple revisions.

Expectations

- You are not expected to consult any other resource, except for the resources that are linked from this page. If you believe that resources are helpful to you and others, consult the instructor.
- Please read and revise your homework at least three times. Note that better writing always requires multiple revisions.
- After you and your partner have completed your revisions, have your partner review your solutions and revise based on the feedback.

Expectations

- You are not expected to consult any other resource, except for the resources that are linked from this page. If you believe that resources are helpful to you and others, consult the instructor.
- Please read and revise your homework at least three times. Note that better writing always requires multiple revisions.
- After you and your partner have completed your revisions, have your partner review your solutions and revise based on the feedback.
- Homework will only be graded to provide feedback. There are no points for the homework.

Expectations

- You are not expected to consult any other resource, except for the resources that are linked from this page. If you believe that resources are helpful to you and others, consult the instructor.
- Please read and revise your homework at least three times. Note that better writing always requires multiple revisions.
- After you and your partner have completed your revisions, have your partner review your solutions and revise based on the feedback.
- Homework will only be graded to provide feedback. There are no points for the homework.
- APS course requires minimal mathematical background. No calculus is required. Knowledge of logarithms will be useful.

Last Name	First Names	Section	Gender	Partner	Mentor	Model	Workshop Mentor
Au-yeung	Nathan	1	M	Rochelle M	AY	Alan Turing	Marco Carmosino
Bennett	Zhaira	1	F	Cody P.	AY	Alan Turing	Marco Carmosino
Gozum	Khelsey	1	F	Karen M.	AY	Alan Turing	Marco Carmosino
Ma	Karen	1	F	Khelsey G.	AY	Alan Turing	Marco Carmosino
Manongdo	Rochelle	1	F	Nathan A.	AY	Alan Turing	Marco Carmosino
Pham	Cody	1	M	Zhaira B.	AY	Alan Turing	Marco Carmosino
Llobrera	Luis	1	M	Martin S.	JC	Hypatia	Daniel Moeller
Mach	Vivian	1	F	Eric X.	JC	Hypatia	Daniel Moeller
Peng	Bruce	1	M	Tiffany W.	JC	Hypatia	Daniel Moeller
Shung	Martin	1	M	Luis L.	JC	Hypatia	Daniel Moeller
Wang	Tiffany	1	F	Bruce P.	JC	Hypatia	Daniel Moeller
Xue	Eric	1	M	Vivian M.	JC	Hypatia	Daniel Moeller
Eldred	Trenten	1	M	Isaac S.	JW	al-Khowarizmi	Nathan Ng
Gobulukoglu	Mustafa	1	M	Arjun S.	JW	al-Khowarizmi	Nathan Ng
Lala	Karan	1	M	Jim L.	JW	al-Khowarizmi	Nathan Ng
Lee	Jim	1	M	Karan L.	JW	al-Khowarizmi	Nathan Ng
Shanley	Isaac	1	M	Trenten E.	JW	al-Khowarizmi	Nathan Ng
Srinivasan	Arjun	1	M	Mustafa G.	JW	al-Khowarizmi	Nathan Ng
Agcaoili	Kristin	1	F	Arvind K.	KL	Ada Lovelace	Mohan Paturi
Crofts	Marian	1	F	Jeremy S.	KL	Ada Lovelace	Mohan Paturi
Kalathil	Arvind	1	M	Kristin A.	KL	Ada Lovelace	Mohan Paturi
Ly	Tyler	1	M	Reese W.	KL	Ada Lovelace	Mohan Paturi
Siocon	Jeremy	1	M	Marian C.	KL	Ada Lovelace	Mohan Paturi
Wahlin	Reese	1	M	Tyler L.	KL	Ada Lovelace	Mohan Paturi
Bayoud	Sinan	1	M	Maya B.	NDS	John von Neumann	Russell Impagliazzo
Bello	Maya	1	F	Sinan B.	NDS	John von Neumann	Russell Impagliazzo
Chen	Warren	1	M	Vachan V.,	NDS	John von Neumann	Russell Impagliazzo
Crow	Nicholas	1	M	Warren C.,	NDS	John von Neumann	Russell Impagliazzo
Diaz-reyes	Kelvin	1	M	Andrew T.	NDS	John von Neumann	Russell Impagliazzo
Lau	Katie	1	F	Nhat N.	NDS	John von Neumann	Russell Impagliazzo
Nguyen	Nhat Quang Hai	1	M	Katie L.	NDS	John von Neumann	Russell Impagliazzo
Tonascia	Andrew	1	M	Kelvin D.	NDS	John von Neumann	Russell Impagliazzo
Vadmal	Vachan	1	M	Warren C.,	NDS	John von Neumann	Russell Impagliazzo

Last Name	First Names	Section	Gender	Partner	Mentor	Model	Workshop Mentor
Kannan	Anish	2	M	Jonathan P	JL	Julia Robinson	Marco Carmosino
Okunev	Sergey Sergeye	2	M	Yacoub O.	JL	Julia Robinson	Marco Carmosino
Oulad Daoud	Yacoub	2	M	Sergey O.	JL	Julia Robinson	Marco Carmosino
Perapalanunt	Jonathan Boon	2	M	Anish K.	JL	Julia Robinson	Marco Carmosino
Shan	David	2	M	Daniel V.	JL	Julia Robinson	Marco Carmosino
Vick	Daniel Allen	2	M	David S.	JL	Julia Robinson	Marco Carmosino
Bontigao	Jacqueline Clau	2	F	Rishabh M.	RD	Leonhard Euler	Daniel Moeller
Juandy	Matthew Albert	2	M	Quoc L.	RD	Leonhard Euler	Daniel Moeller
Luong	Quoc Toan Min	2	M	Matthew J.	RD	Leonhard Euler	Daniel Moeller
Malhotra	Rishabh	2	M	Jacqueline	RD	Leonhard Euler	Daniel Moeller
Phan	Khoi Minh	2	M	June S.	RD	Leonhard Euler	Daniel Moeller
Suk	June Young	2	F	Khoi P.	RD	Leonhard Euler	Daniel Moeller
Bach	Triet Minh	2	M	Alicia C.	ST	Charles Babbage	Nathan Ng
Chen	Alicia Miao	2	F	Triet B.	ST	Charles Babbage	Nathan Ng
Lee	Pearl Jinjoo	2	F	Sherman L.	ST	Charles Babbage	Nathan Ng
Lee	Sherman Henry	2	M	Pearl L.	ST	Charles Babbage	Nathan Ng
Smith	Jeremy Seiji	2	M	Aaron Y.	ST	Charles Babbage	Nathan Ng
Yang	Aaron	2	M	Jeremy S.	ST	Charles Babbage	Nathan Ng
Chou	Vanessa	2	F	Jed T.	VC	Leelavati	Mohan Paturi
Ho	Thong Ung	2	M	Ethan V.	VC	Leelavati	Mohan Paturi
Ponce	Nancy Jacqueli	2	F	Connor S.	VC	Leelavati	Mohan Paturi
Seong	Daniel Shinyup	2	M	Christine W	VC	Leelavati	Mohan Paturi
Smith	Connor A	2	M	Nancy P.	VC	Leelavati	Mohan Paturi
Tadios	Jed Matthew	2	M	Vanessa C.	VC	Leelavati	Mohan Paturi
Vander Horn	Ethan Douglas	2	M	Thong H.	VC	Leelavati	Mohan Paturi
Wu	Christine May	2	F	Daniel S.	VC	Leelavati	Mohan Paturi
Chen	Julia	2	F	Guiseppie	YEW	Qin Jiushao	Russell Impagliazzo
Huang	Tiffany Mae	2	F	Krishi V.	YEW	Qin Jiushao	Russell Impagliazzo
Justo	David Antonio	2	M	Jenny W.	YEW	Qin Jiushao	Russell Impagliazzo
Pagliaro	Guiseppie Robe	2	M	Julia C.	YEW	Qin Jiushao	Russell Impagliazzo
Vemula	Krishi Kesav	2	M	Tiffany H.	YEW	Qin Jiushao	Russell Impagliazzo
Wong	Jenny	2	F	David J.	YEW	Qin Jiushao	Russell Impagliazzo

Search Problems

- Problems come in the form of searching for an object in a set.

Search Problems

- Problems come in the form of searching for an object in a set.
- Explicitly identify the set of possible scenarios.

Search Problems

- Problems come in the form of searching for an object in a set.
- Explicitly identify the set of possible scenarios.
- Typical search algorithm: A series of steps where in each step we
  - perform some computation, and
  - ask a question to narrow down the possibilities.

Search Problems

- Problems come in the form of searching for an object in a set.
- Explicitly identify the set of possible scenarios.
- Typical search algorithm: A series of steps where in each step we
  - perform some computation, and
  - ask a question to narrow down the possibilities.
- Each possible answer to the question reduces the problem to a subproblem where we have fewer possibilities.

Search Problems

- Problems come in the form of searching for an object in a set.
- Explicitly identify the set of possible scenarios.
- Typical search algorithm: A series of steps where in each step we
  - perform some computation, and
  - ask a question to narrow down the possibilities.
- Each possible answer to the question reduces the problem to a subproblem where we have fewer possibilities.
- The set of possible scenarios is partitioned along the possible answers to the question.

Search Problems

- Problems come in the form of searching for an object in a set.
- Explicitly identify the set of possible scenarios.
- Typical search algorithm: A series of steps where in each step we
  - perform some computation, and
  - ask a question to narrow down the possibilities.
- Each possible answer to the question reduces the problem to a subproblem where we have fewer possibilities.
- The set of possible scenarios is partitioned along the possible answers to the question.
- Design questions so that the number of possibilities in each case is as equal as it can be to the number of possibilities in other cases.

A Fake among Eight Coins

- There are eight identical-looking coins; one of these coins is counterfeit and is known to be lighter than the genuine coins.

A Fake among Eight Coins

- There are eight identical-looking coins; one of these coins is counterfeit and is known to be lighter than the genuine coins.
- What is the minimum number of weighings needed to identify the fake coin with a two-pan balance scale without weights?

A Fake among Eight Coins: Solution

- Divide the coins into three groups of 3, 3, and 2 coins each

A Fake among Eight Coins: Solution

- Divide the coins into three groups of 3, 3, and 2 coins each
- Call them L, R, and E respectively.

A Fake among Eight Coins: Solution

- Divide the coins into three groups of 3, 3, and 2 coins each
- Call them L, R, and E respectively.
- Use the two-pan balance to compare the weights of the groups L and R.

A Fake among Eight Coins: Solution

- Divide the coins into three groups of 3, 3, and 2 coins each
- Call them L, R, and E respectively.
- Use the two-pan balance to compare the weights of the groups L and R.
- If they weigh the same, the fake coin can neither be in L nor in R, so it must be in E.

A Fake among Eight Coins: Solution

- Divide the coins into three groups of 3, 3, and 2 coins each
- Call them L, R, and E respectively.
- Use the two-pan balance to compare the weights of the groups L and R.
- If they weigh the same, the fake coin can neither be in L nor in R, so it must be in E.
- Take the two coins in E and compare them. The lighter coin (between the two coins in E) must be the fake coin.

A Fake among Eight Coins: Solution

- If the weights of L and R are different, then the fake coin must be in the group that is lighter.

A Fake among Eight Coins: Solution

- If the weights of L and R are different, then the fake coin must be in the group that is lighter.
- Take any 2 coins from this group and weigh them.

A Fake among Eight Coins: Solution

- If the weights of L and R are different, then the fake coin must be in the group that is lighter.
- Take any 2 coins from this group and weigh them.
- If they weigh the same then the remaining coin in the group must be fake.

A Fake among Eight Coins: Solution

- If the weights of L and R are different, then the fake coin must be in the group that is lighter.
- Take any 2 coins from this group and weigh them.
- If they weigh the same then the remaining coin in the group must be fake.
- If one of them is lighter, the lighter coin must be the fake coin.

A Fake among Eight Coins: Solution

- If the weights of L and R are different, then the fake coin must be in the group that is lighter.
- Take any 2 coins from this group and weigh them.
- If they weigh the same then the remaining coin in the group must be fake.
- If one of them is lighter, the lighter coin must be the fake coin.
- In all cases, we identified the counterfeit with two comparisons.

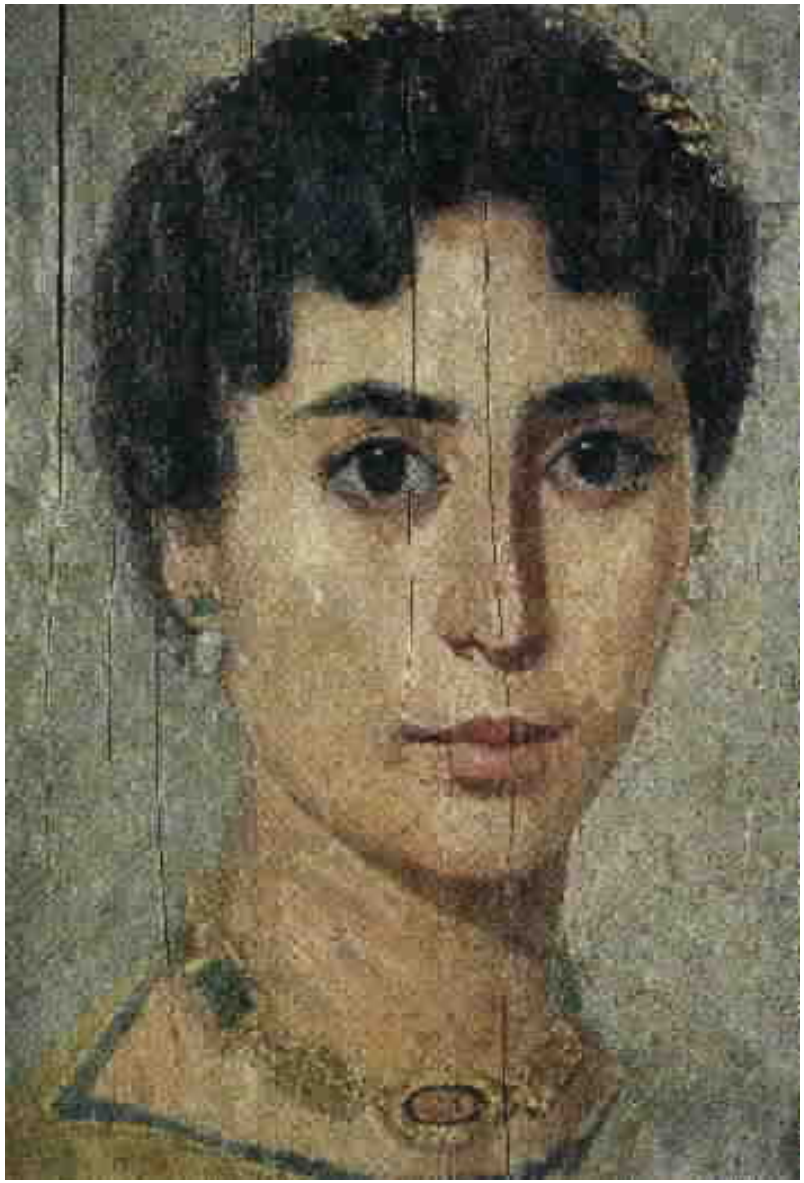


Figure: Hypatia

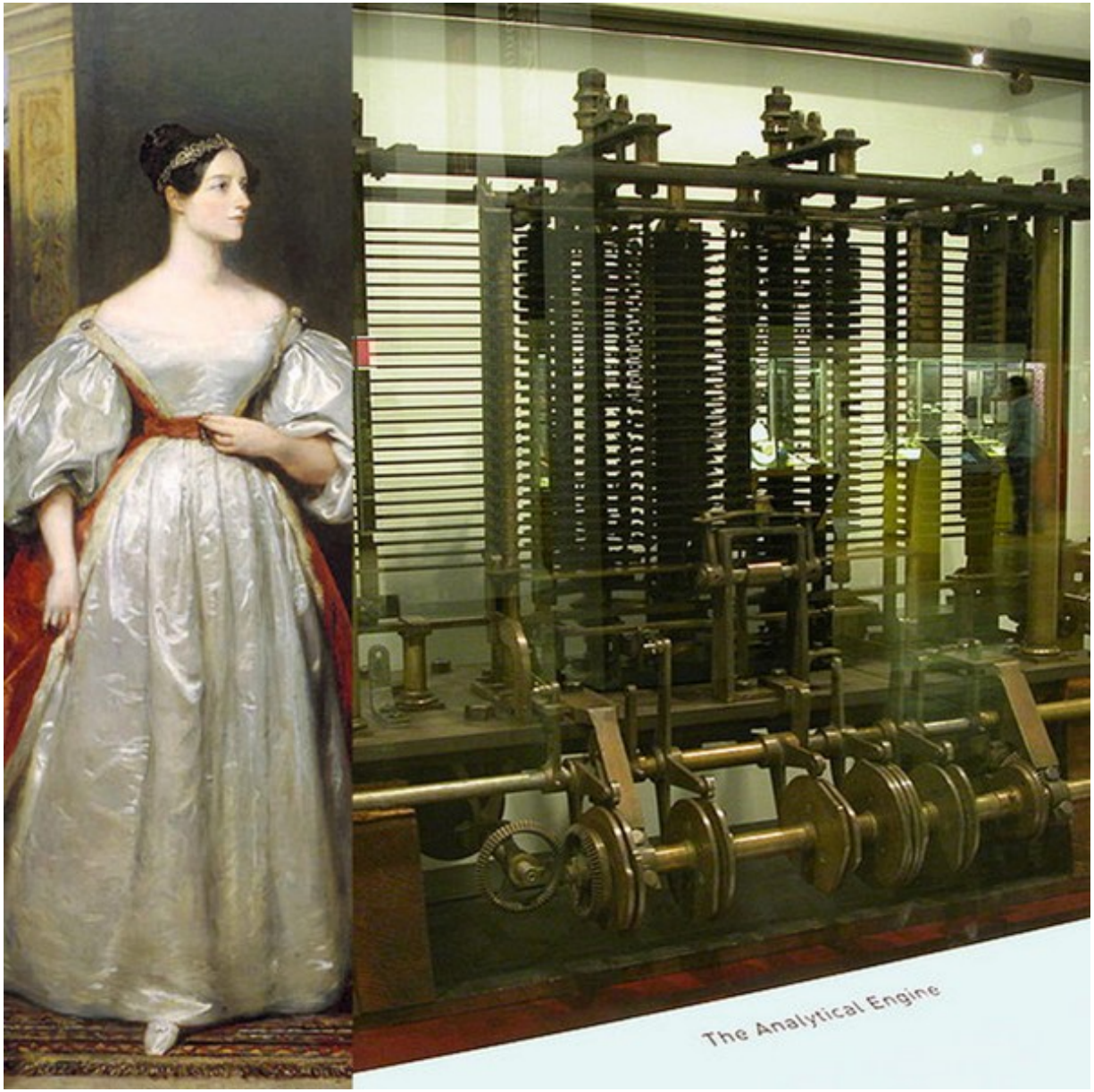


Figure: Ada Lovelace

Muhammad ibn Musa al-Khwarizmi

- Muhammad ibn Musa al-Khwarizmi (approx 800-847 CE)
- Born around 800 CE in Khwarizm, now in Uzbekistan.
- Al-Khwarizmi lived in Baghdad, where he worked at the "House of Wisdom" (*Dar al-Hikma*) under the Caliph Harun al-Rashid.
- What is the origin of the word Algorithm?

Fundamentals of Software Development 1

Slide 1

Figure: al-Khowarizmi

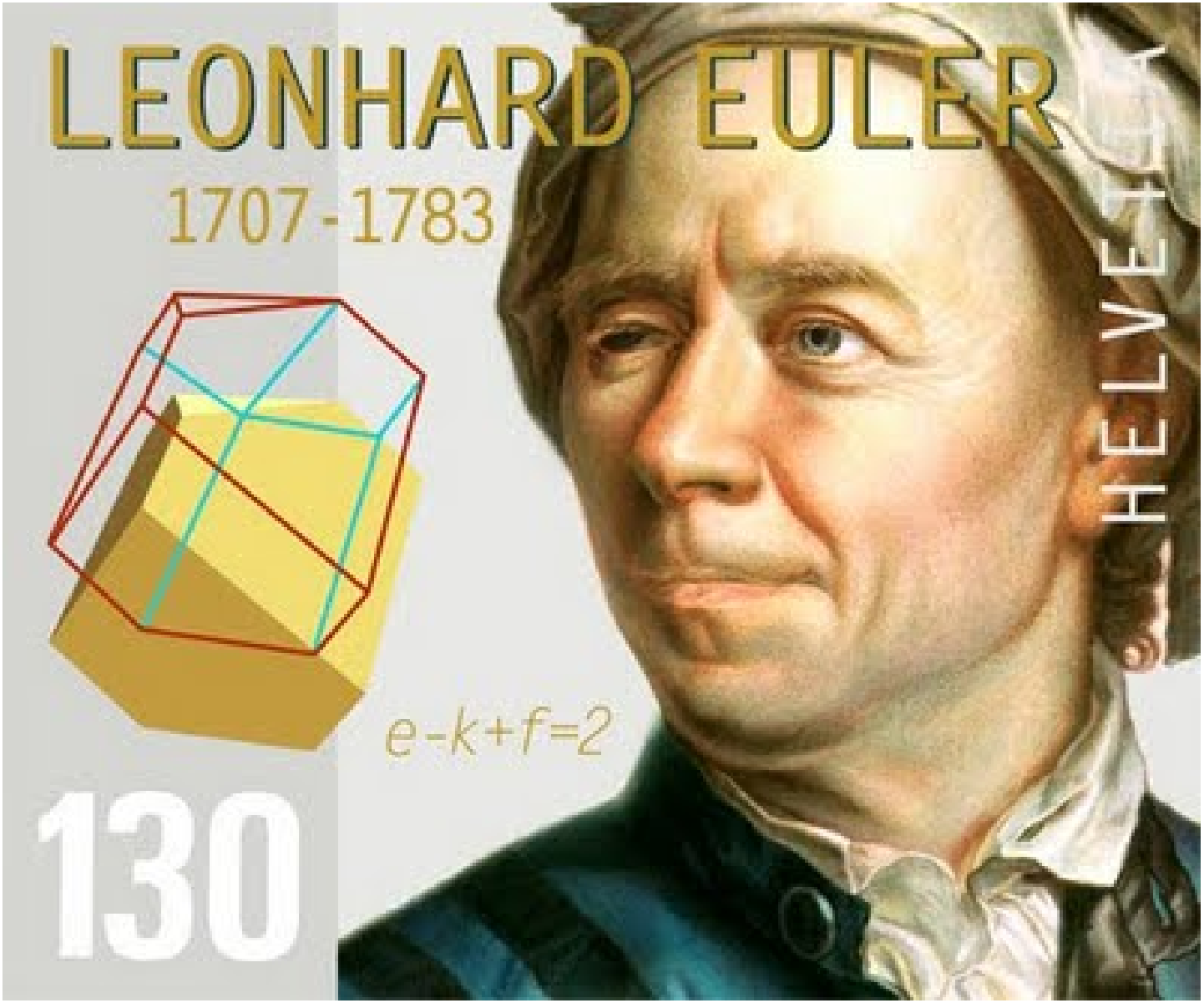


Figure: Leonhard Euler

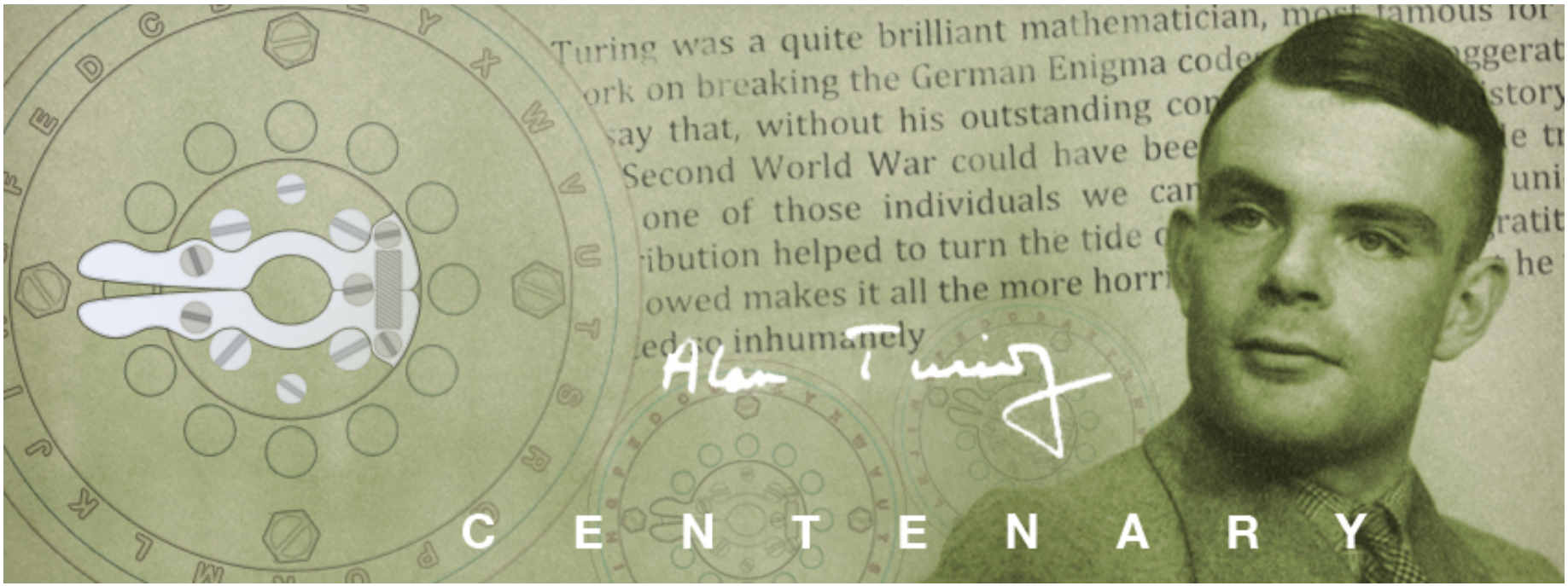


Figure: Alan Turing

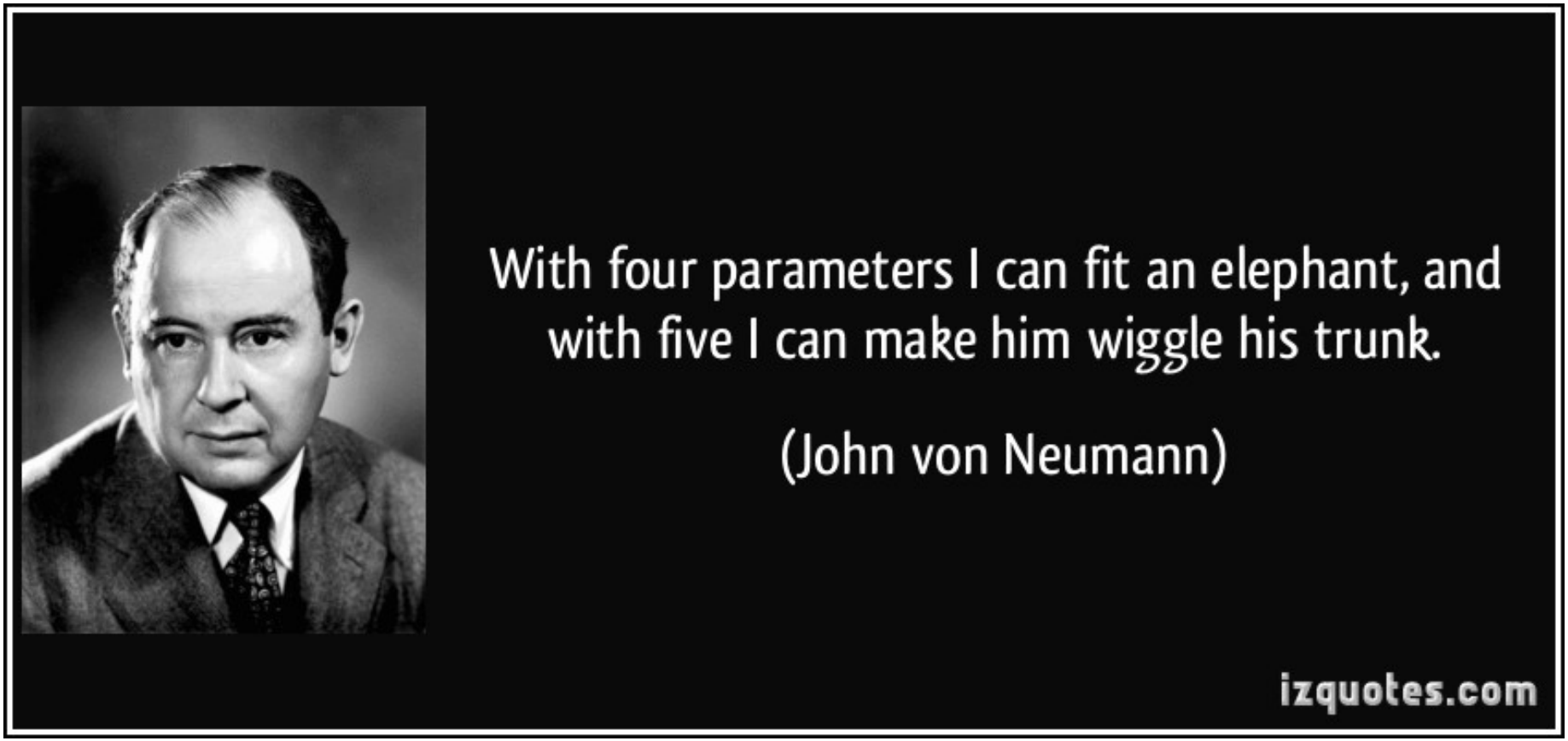


Figure: John von Neumann


Co-Sponsored by the Mathematical Sciences Research Institute and the UC Berkeley Department of Mathematics
Julia Robinson
And Hilbert's Tenth Problem
A film by
George Csicsery
Wednesday, April 30, 2008
7pm to 9pm
Room 2050 (Chan Shun Auditorium)
in the Valley Life Sciences Building
at UC Berkeley
Post-screening panel discussion
with Constance Reid (sister and
biographer of Julia Robinson),
filmmaker George Csicsery, and
mathematicians Martin Davis,
Dana Scott and Bjorn Poonen.
Moderated by Alan Weinstein,
UCB Math Dept. Chair.
The story of an American mathematician
and her passionate pursuit and triumph
over an unsolved problem.
Hilbert's 10th Problem (1900): Is there an algorithm for
deciding whether a polynomial equation with integer
coefficients has an integer solution?

Figure: Charles Babbage

Figure: Qin Jiushao

Figure: Bhaskara II (Leelavati is Bhaskara's daughter and the name of his book)

Thank You