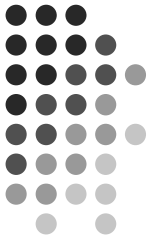
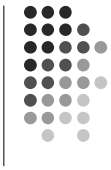
Teaching the Principles of Computation

Tom Cortina
Carnegie Mellon University



1

Introducing CS to students



- What is Computer Science (CS)?
  - Students see CS as computer programming.
    - "The more languages you know, the better you are as a computer scientist."
    - "The more you know about one language, the better you are as a computer scientist."
    - "All computer scientists write Java programs."
- If students are going to take only one course introducing them to computer science, how do we show them that there's more to CS than Java programming?

2

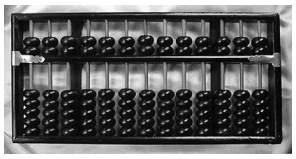
History of Computing

- What historical events brought us to the modern computer?
- When did the term *computer* originate?

The Oxford English Dictionary still describes a computer as
"a person employed to make calculations in an observatory, in surveying, etc."
(atariarchives.org)

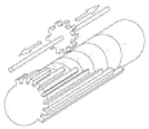
3

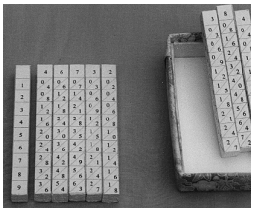
The Early Days of Computing



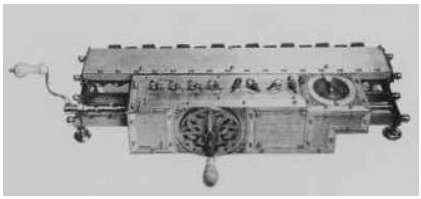
Abacus (500 BC)

Pascaline (1643)





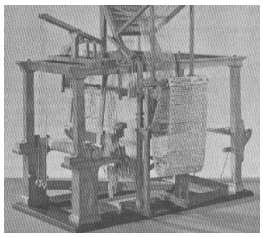
Napier's Bones (1617)



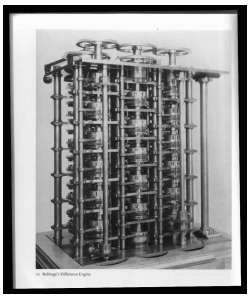
Leibniz' step reckoner (1674)

4

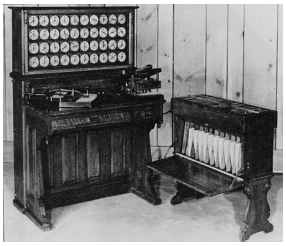
Mechanical Computing



Jacquard's Loom (1801)



Charles Babbage's Difference Engine (1832) (computed finite differences)



Hollerith's Census Collator (1890)

5

Finite Differences

$F(x) = x^2 + 3x + 6$

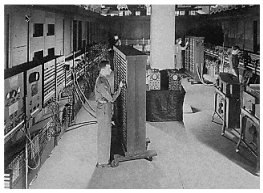
$\Delta F(x) = F(x+1)-F(x) = 2x + 4$

$\Delta^2 F(x) = \Delta F(x+1) - \Delta F(x) = 2$

x	$\Delta^2 F(x)$	$\Delta F(x)$	F(x)
0	2	4	6
1	2	6	10
2	2	8	16
3	2	10	24
…			

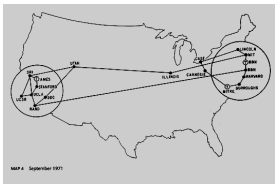
6

Electronic Computing



ENIAC
(1946)

UNIVAC I
(1951)



ARPANET

Personal Computer
(1981)

The Creation of the Computer
(History Channel)

Faces of Computing

20 Sergey Brin

Algorithms:
Specifying Computation

- An algorithm is “a precise rule (or set of rules) specifying how to solve some problem.”
  (thefreedictionary.com)
- Algorithms in real life:
  - Recipes
  - Completing a Tax Form
  - Knitting
- Have students go out and find algorithms in their everyday lives!

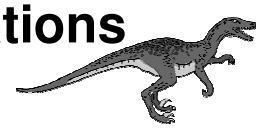
9

GCD Algorithm

- Attributed to the Greek mathematician Euclid between 400 and 300 B.C.
- Algorithm to find the GCD of x and y:
  1. Input x
  2. Input y
  3. While y is not 0, do the following:
     a. Set x' equal to y
     b. Set y' equal to x modulo y
     c. Set x = x' and y = y'
  4. Output x as the GCD

10

Visualizing Computations using RAPTOR



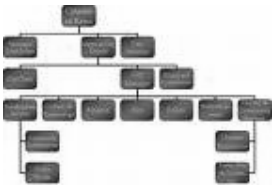
- Developed by Martin Carlisle and others at the US Air Force Academy.
- Uses flowcharts to allow students to build computations and simulate them with almost no syntax.
- Includes support for input, conditionals, loops, subroutines, graphics, sound, text output

Google: Raptor Martin Carlisle

11

Data Structures

- Arrays (Vectors, Matrices)
- Linked Lists
- Stacks
- Queues
- (Binary) Trees
- Graphs

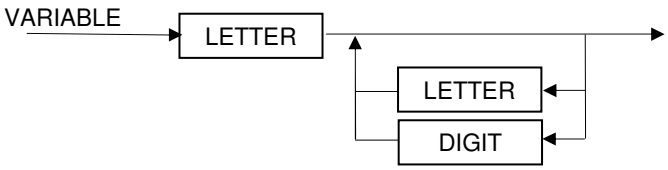


12

Expressing Syntax

- Computer scientists have invented various ways to explicitly describe complex rules of syntax.
  - A valid variable name is a string that starts with a letter and is followed by any number of letters or digits in any combination.

Syntax diagram



VARIABLE ::= LETTER { LETTER | DIGIT } EBNF

13

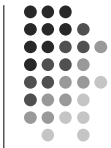
Programming Languages

- Why are there so many programming languages?

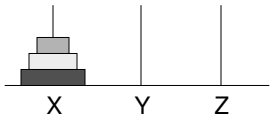
Google: Gunnerson Programming Languages
Google: O'Reilly History Programming

14

Recursion & Algorithmic Techniques



- Students learn that computer science effectively uses recursion to express solutions to complex problems in a very concise manner.



Towers of Hanoi

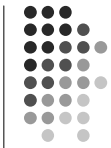
How many moves to solve this puzzle for n discs?

$2^n - 1$

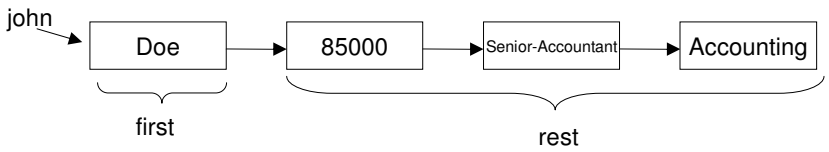
For n = 64, if the monks move 1 disc/sec, it would take them about 585 billion years to complete!

15

Recursion in Scheme



```
(define john (list 'Doe 85000
     'Senior-Accountant 'Accounting))
(define jane (list 'Smith 97000
     'Manager 'Web-Services))
(define (salary employee)
  (first (rest employee)))
(salary john)
85000
```

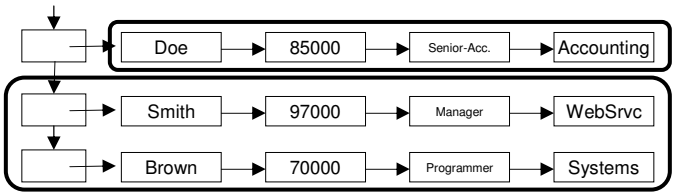


16

Recursion in Scheme (cont'd)

```
(define (sum-salaries employees)
  (if (null? employees)
      0
      (+ (salary (first employees))
         (sum-salaries (rest employees)))))

(sum-salaries (list john jane mike))
```



17

Correctness

- How do we know if a computation is correct? Why is this important?


Mariner I


Therac-25


AT&T Bug


Scud Missiles

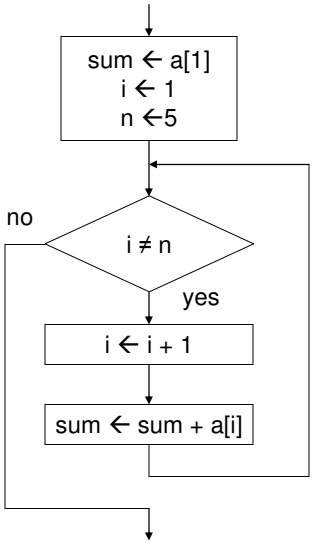

Denver Airport

- Showing correctness:
  - Loop invariants
  - Induction

Google: AT&T Phone Crash 1990
Google: Patriot Missile Failure

18

Loop to add up all data in array



Invariant:

$$sum = \sum_{x=1}^{i} a[x]$$

a

1	2	3	4	5
10	20	30	40	50

i	1	2	3	4	5
sum	10	30	60	100	150

19

Efficiency

- Assume an algorithm requires N data values to process. If each operation takes 1 μs (1 millionth of a second) to execute.
- How many μs will it take to run the algorithm on N=100 data values if the algorithm has the following number of operations?
  N, N $\log_2$N, N^2, N^3, N^4, 2^N, N!, N^N

20

Comparing Algorithms (cont'd)

Number of Computations	Execution Time
N	100 µs
$N \log_2 N$	665 µs
N^2	10,000 µs = 0.01 sec
N^3	1,000,000 µs = 1 sec
N^4	100,000,000 µs = 1 min 40 sec
2^N	> 10^{30} µs = 3.5 quintillion centuries!
$N!$	> 10^{160} µs
N^N	> 10^{201} µs

Number of data values
N = 100

The number of protons in the known universe is < 10^{79}
The number of microseconds since the Big Bang is < 10^{24}.

21

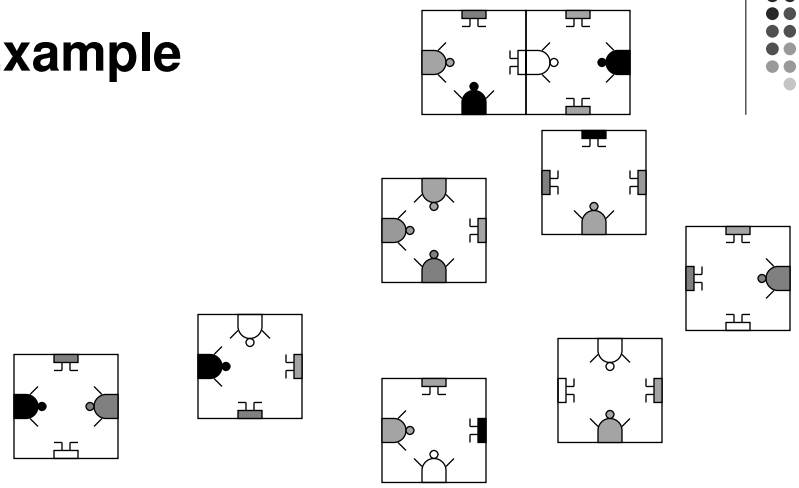
Monkey Puzzle Problem

- Given:
  - A set of N square cards whose sides are imprinted with the upper and lower halves of colored monkeys.
  - N is a square number, such that $N = M^2$.
  - Cards cannot be rotated.
- Problem:
  - Determine if an arrangement of the N cards in an M X M grid exists such that each adjacent pair of cards display the upper and lower half of a monkey of the same color.

Source: www.dwheeler.com (2002)

22

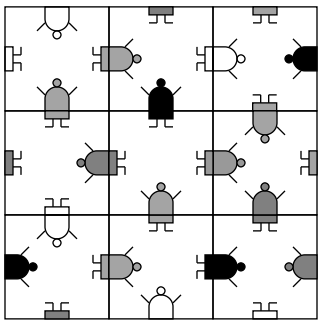
Example



Images from: **Simonas Šaltenis,** Aalborg University, simas@cs.auc.dk

23

Analysis



If there are N = 9 cards (M = 3):

To fill the first cell, we have 9 card choices.
To fill the second cell, we have 8 card choices left.
To fill the third cell, we have 7 card choices remaining.
etc.

The total number of unique arrangements for N = 9 cards is:
9 x 8 x 7 x 6 x 5 x 4 x 3 x 2 x 1 = 9! = 362,880

The total number of touching cards we need to check for N = 9 cards is 12.

24

Intractability

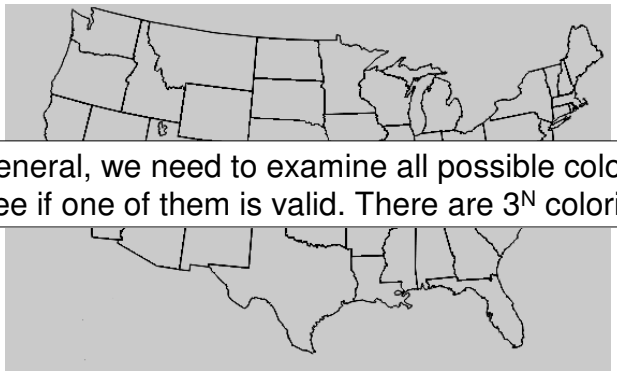
For N cards, the number of arrangements to examine is O(N!)

If we can analyze one arrangement in a millisecond:

N	Time to analyze all arrangements
9	362,880 ms ≈ 363 s ≈ 6 min
16	20,922,789,888,000 ms ≈ 663 years
25	15,511,210,043,330,985,984,000,000 ms ≈ 500 quadrillion centuries

25

Map Coloring

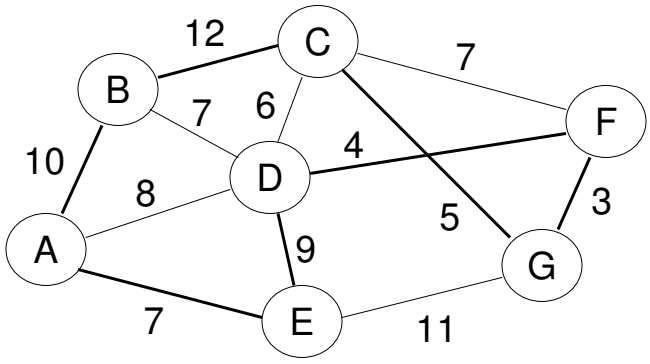


In general, we need to examine all possible colorings to see if one of them is valid. There are 3^N colorings!

Given a map of N territories, can the map be colored using 3 colors such that no two adjacent territories are colored with the same color?

26

Traveling Salesperson



Is there a route with cost at most 52?
Is there a route with cost at most 48?

In general, we need to compute all possible routes. There are (N-1)! possible routes in the worst case.

27

Computability

- The class **P** consists of all those decision problems that can be solved on a deterministic sequential machine (i.e. a computer) in an amount of time that is polynomial in the size of the input
- The class **NP** consists of all those decision problems whose positive solutions can be verified in polynomial time.
- The Clay Mathematics Institute is offering a $1M prize for the first person to prove P = NP or P ≠ NP. (http://www.claymath.org/millennium/P_vs_NP/)

28

Undecidable Problems

- Can we write a computer program that determines if any computer program halts when given a specific input?
- Turing's famous Halting Problem
- Students are surprised to find out that the answer is NO!
  - There are some problems that a computer cannot solve no matter how much power or time it has.

29

Halting Problem

Assume such a program H exists to determine if a program halts when given some input.

H(*p*, *q*) returns true if program *p* halts given input *q*.

H(*p*, *q*) returns false if program *p* does not halt given input *q*.

Define program S as follows:

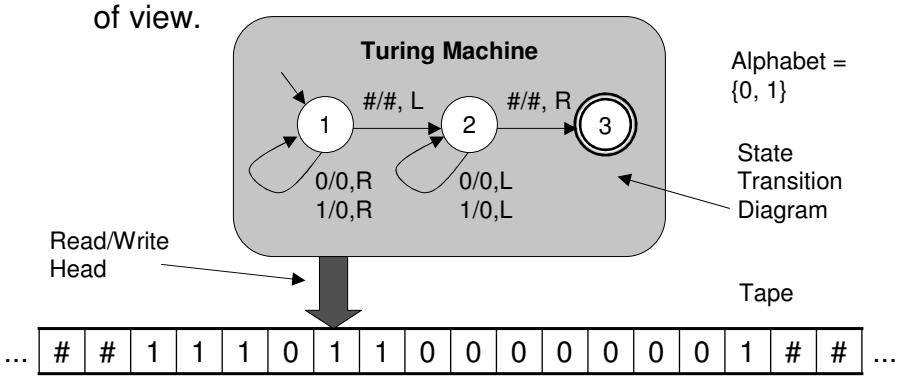
```
program S:
    input program p
    if (H(p,p)) loop forever
    else stop
```

What happens when we run S with itself as input?

30

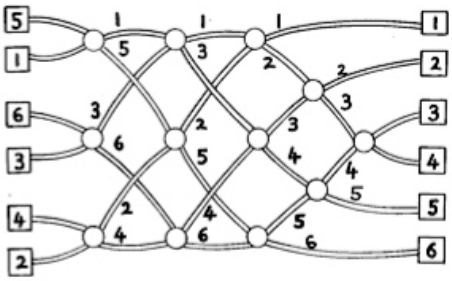
Turing Machines

- Introduces state diagrams to students.
- Students look at computation from a "hardware" point of view.



Working in Parallel

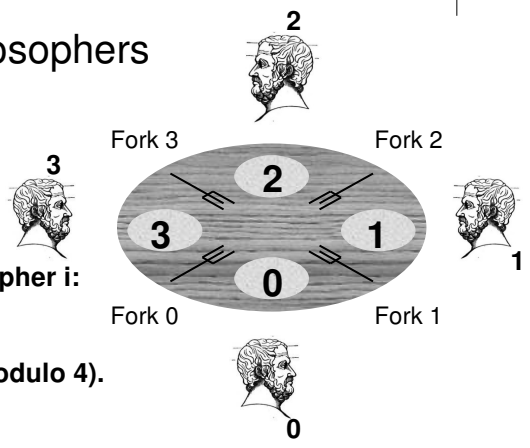
- Sorting 6 numbers sequentially can take up to 15 steps. (bubble sort)
- A parallel network can do it in just 5 steps.



from *Computer Science Unplugged*

Deadlock

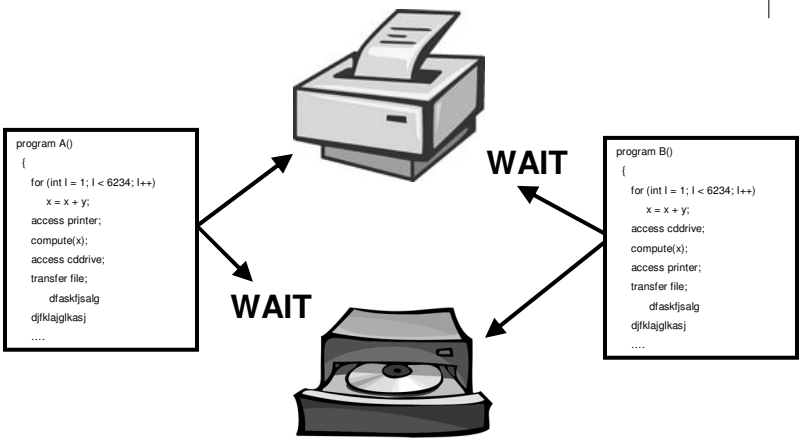
- The Dining Philosophers Problem



Algorithm for Philosopher i:
1. THINK
2. Pick up fork i.
3. Pick up fork i+1 (modulo 4).
4. EAT
5. Put down fork i.
6. Put down fork i+1 (modulo 4).
7. Go to step 1.

33

Deadlock in a computer



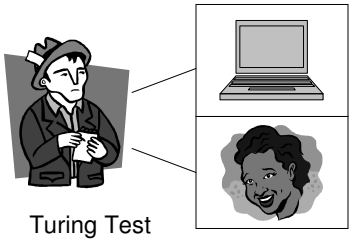
The design of operating systems includes algorithms to avoid deadlock.

34

Artificial Intelligence

- What does it mean for a machine to be intelligent?
- Allen Newell and Herbert Simon (from CMU) contributed to one of the first AI programs, the General Problem Solver (GPS) in 1957.
- Turing Test

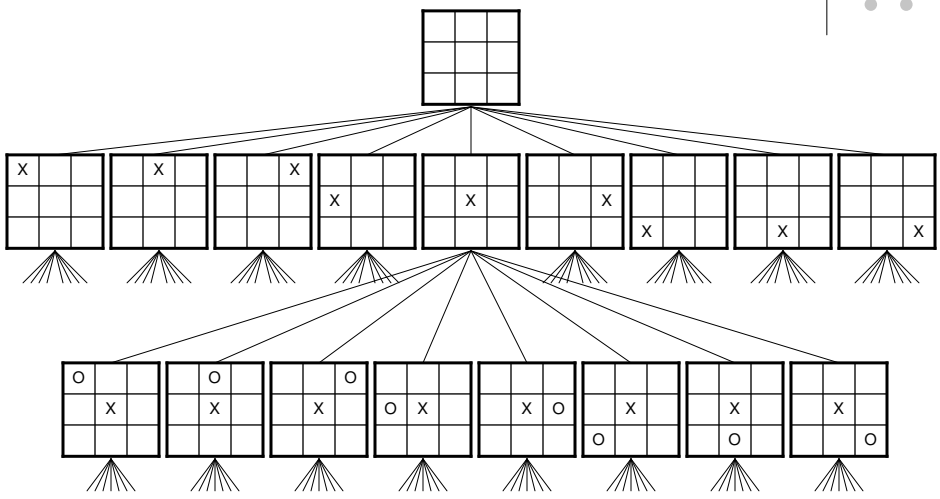
Google: Loebner Prize AI



Turing Test

35

Tic-Tac-Toe: Game Tree



36

Tic Tac Toe

- How many configurations are on the 1st level of the tree (below the root)?
- How many configurations are on the 2nd level of the tree?
- If we were forced to play the game until all 9 squares were filled, how many leaves would the tree have?
- Game trees are usually extremely large, so game programs also use heuristics to narrow the search space in the tree that must be examined.

37

Deep Blue

- IBM's "Deep Blue" computer beats Gary Kasparov in a chess match in 1997.
- Heuristics values:
  - The value of each piece. (1 for pawn up to 9 for queen)
  - The amount of control each side has over the board.
  - The safety of the king.
  - The quickness that pieces move into fighting position.
- Is Deep Blue intelligent?
- What's the next AI challenge?

38

The future of computing

NANOCOMPUTING

Quantum Computing

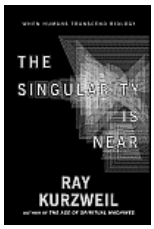
DNA computing

39

Futuristic Views

Robot: Mere Machine to Transcendent Mind
Hans Moravec

The Future of Ideas
Lawrence Lessig



The Singularity is Near
Ray Kurzweil

Flesh and Machines
Rodney A. Brooks

Nanofuture
J. Storrs Hall

40

CS ≠ Java

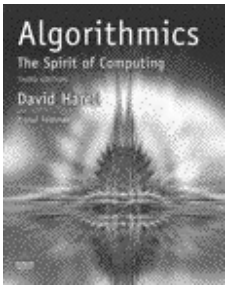
- We can show students that computer science has much more depth than just learning how to program in Java.
- Try to insert a few ideas illustrating computer science into your AP class.
- Build a new course as an intro to the AP course that illustrates computational thinking.

41

For More Info

- 15-105 Principles of Computation
- **http://www.cs.cmu.edu/~tcortina/15-105**

Google: 15-105



Algorithmics: The Spirit of Computing (3rd Edition)
by David Harel with Yishai Feldman
Publisher: Addison-Wesley (2004)
ISBN: 0321117840

42