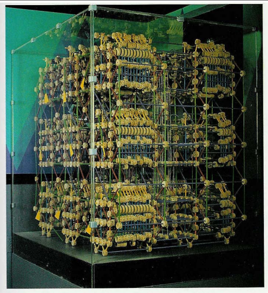
Class 33: Computing with Photons and Life



Color Plate 1 Early model of the Tinkertoy computer, on display at The Computer Museum, Boston, also plays tic-tac-toe.

From *The Tinkertoy Computer and Other Machinations* by A. K. Dewdney
http://www.amazon.com/exec/obidos/tg/detail/-/071672491X/103-4408705-5367831?v=glance

CS150: Computer Science
University of Virginia
Computer Science

David Evans
http://www.cs.virginia.edu/evans

Church-Turing Thesis

- Church's original (1935)
  - Lambda calculus is equivalent to real world computers (can compute any computable function)
- Turing's version
  - "Every function which would naturally be regarded as computable can be computed by a Turing machine."
- Generalized version:
  - Any computation that can be done by an algorithm can be done by a mechanical computer
  - All "normal" computers are equivalent in computing power

Turing Machines and Complexity

- Stronger version:
  - Complexity classes P, NP, and NP-complete are defined for Turing machine steps, but apply identically to all "normal" computers
- Today: "Abnormal" Computers
  - *Might* change what is computable (probably don't)
  - Do change what a normal "step" is

Normal Steps

- Turing machine:
  - Read one square on tape, follow one FSM transition rule, write one square on tape, move tape head one square
- Lambda calculus:
  - One beta reduction
- Your PC:
  - Execute one instruction (?)
    - What one instruction does varies

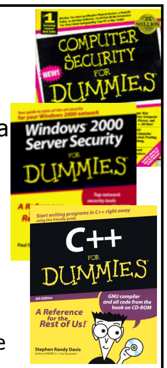
Generalized Normal Steps

- Require a constant amount of time
- Perform a fixed amount of work
  - Localized
  - Cannot scale (indefinitely) with input size

Abnormal Imaginary Computer

- "Accelerating" TM
  - Like a regular TM, except the first step takes 1 second, second step takes ½ second, third step takes ¼ second, ... n^{th} step takes $1/2^n$ second
- Is our "Accelerating" TM more powerful than a regular TM?

Quantum Physics for Dummies



- Light behaves like both a wave and a particle at the same time
- A single photon is in many states at once
- Can't observe its state without forcing it into one state
- Schrödinger's Cat
  - Put a live cat in a box with cyanide vial that opens depending on quantum state
  - Cat is both dead and alive at the same time until you open the box

Quantum Computing

- Feynman, 1982
- Quantum particles are in all possible states
- Can try lots of possible computations at once with the same particles
- In theory, can test all possible factorizations/keys/paths/etc. and get the right one!
- In practice, very hard to keep states entangled: once disturbed, must be in just one possible state

Qubit

- Regular bit: either a 0 or a 1
- Quantum bit: 0, 1 or in between
  - p% probability it is a 1
- A single qubit is in 2 possible states at once
- If you have 7 bits, you can represent any one of 2^7 different states
- If you have 7 qubits, you have 2^7 different states (at once!)

Quantum Computers Today

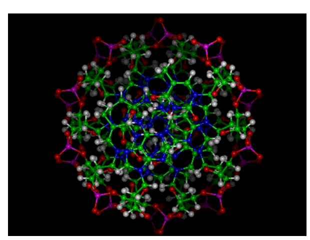
- Several quantum algorithms
  - Shor's algorithm: factoring using a quantum computer
- Actual quantum computers
  - 5-qubit computer built by IBM (2001)
  - Implemented Shor's algorithm to factor: 15 (= 5 * 3)
    - "World's most complex quantum computation"
  - Los Alamos has built a 7-qubit computer
- To exceed practical normal computing need > 30 qubits
  - Adding another qubit is more than twice as

Complexity for Quantum Computer

- Complexity classes are different than for regular computers, because a step is different
- Quantum computer: each step can take both possible decisions at once
  - Means a quantum computer is a nondeterministic computer!
  - It can solve problems in class NP in polynomial time!
- What matters?

Number of qubits you need
Number of (nondeterministic) steps

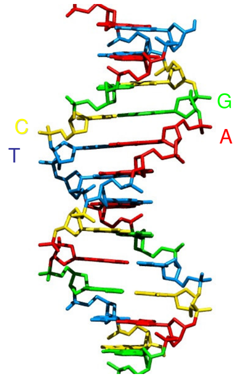
Computing with DNA



Leonard Adleman (Mathematical Consultant for *Sneakers*), 1995

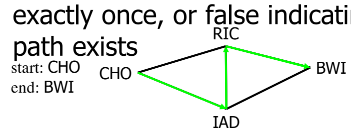
DNA

- Sequence of nucleotides: adenine (A), guanine (G), cytosine (C), and thymine (T)
- Two strands, A must attach to T and G must attach to C



Hamiltonian Path Problem

- Input: a graph, start vertex and end vertex
- Output: either a path from start to end that touches each vertex in the graph exactly once, or false indicating no such path exists



How hard is the Hamiltonian path problem?

Encoding The Graph

- Make up a two random 4-nucleotide sequences for each city:

CHO:	CHO_1 = ACTT	CHO_2 = gcag
RIC:	RIC_1 = TCGG	RIC_2 = actg
IAD:	IAD_1 = GGCT	IAD_2 = atgt
BWI:	BWI_1 = GATC	BWI_2 = tcca

- If there is a link between two cities (A→B), create a nucleotide sequence: A_2B_1

CHO→RIC gcagTCGG
RIC→CHO actgACTT

Based on Fred Hapgood's notes on Adelman's talk
http://www.mitre.org/research/nanotech/hapgood_on_dna.html

Encoding The Problem

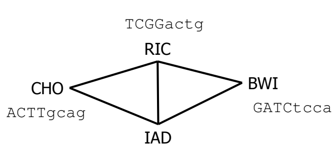
- Each city nucleotide sequence binds with its complement (A ↔ T, G ↔ C) :

CHO:	CHO_1 = ACTT	CHO_2 = gcag
CHO':	TGAA	cgtc
RIC:	TCGGactg	
RIC':	AGCCtgac	
IAD:	GGCTatgt	IAD' = CCGAtaca
BWI:	GATCtcca	BWI' = CTAGaggt

- Mix up all the link and complement DNA strands – they will bind to show a path!

Path Binding

CHO' IAD' RIC' BWI'
TGAAcgtcCCGAtacaAGCCtgacCTAGaggt
gcagGGCTatgtTCGGactgGATC
CHO→IAD IAD→RIC RIC→BWI



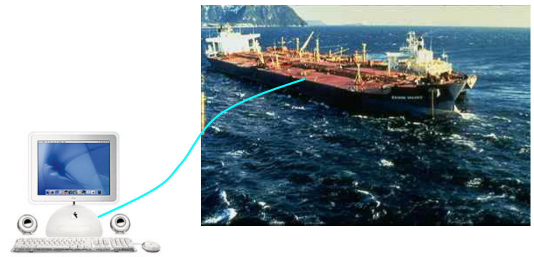
Getting the Solution

- Extract DNA strands starting with CHO and ending with BWI
  - Easy way is to remove all strands that do not start with CHO, and then remove all strands that do not end with BWI
- Measure remaining strands to find ones with the right weight (7 * 8 nucleotides)
- Read the sequence from one of these strands

Why don't we use DNA computers?

- Speed: shaking up the DNA strands does 10^{14} operations per second ($400M supercomputer does 10^{10})
- Memory: we can store information in DNA at 1 bit per cubic nanometer
- How much DNA would you need?
  - Volume of DNA needed grows exponentially with input size
  - To solve ~45 vertices, you need ~20M gallons

DNA-Enhanced PC



Complexity for DNA Computer

- Complexity classes are different than for regular computers, because a step is different
  - Scales with amount of DNA you have
- Turns time complexity into "DNA-amount" complexity
  - If you have an amount of DNA that is exponential in input size, you can solve NP complete problems in P steps!

Computability for Quantum and DNA computers

- DNA computers: no change to what is computable, only changes time it takes
- Quantum computers:
  - They are so strange they even change what is computable!
  - Quantum physics provides true randomness, something a Turing machine cannot do

Charge

- Exam 2 out Friday
  - Covers through Monday
    - No questions on Quantum physics or DNA
  - Links to example exams on the web
  - Review session Wednesday, 7pm