Introduction to Computer Science

CS A101

What is Computer Science?

- First, some misconceptions.
- **Misconception 1**: I can put together my own PC, am good with Windows, and can surf the net with ease, so I know CS.
- **Misconception 2**: Computer science is the study of how to write computer programs.
- **Misconception 3**: Computer science is the study of the uses and applications of computers and software.

Computer Science

- **Computer science is the study of algorithms, including**
  - Their formal and mathematical properties
  - Their hardware realizations
  - Their linguistic realizations
  - Their applications

What Will We Cover?

- Broad survey of computer science topics, some depth in programming, more on breadth
- Topics
  - History
  - Data representation
  - Computer architecture (software perspective)
  - Operating Systems
  - Networking
  - Algorithms
  - Theory
  - Database Systems
  - Programming (more depth than other topics)

Terminology

- **Algorithm:** A set of steps that defines how a task is performed
- **Program:** A representation of an algorithm
- **Programming:** The process of developing a program
- **Software:** Programs and algorithms
- **Hardware:** Physical equipment

History of Algorithms

- The study of algorithms was originally a subject in mathematics.
- Early examples of algorithms
  - Long division algorithm
  - Euclidean Algorithm
- **Gödel's Incompleteness Theorem**: Some problems cannot be solved by algorithms.

Example: Euclid's algorithm

Description: This algorithm assumes that its input consists of two positive integers and proceeds to compute the greatest common divisor of these two values.

Procedure:

Step 1. Assign M and N the value of the larger and smaller of the two input values, respectively.

Step 2. Divide M by N, and call the remainder R.

Step 3. If R is not 0, then assign M the value of N, assign N the value of R, and return to step 2; otherwise, the greatest common divisor is the value currently assigned to N.

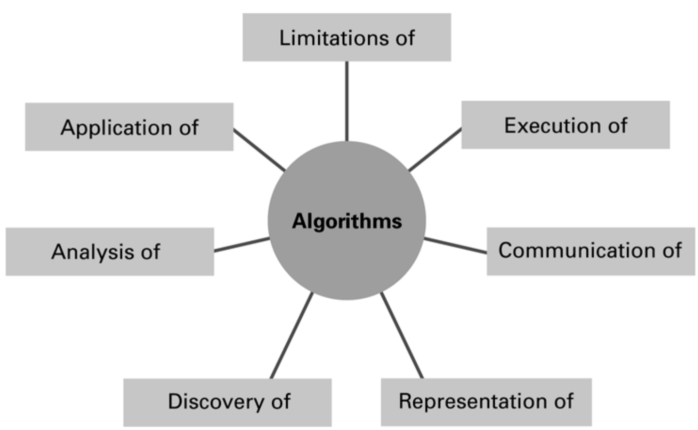
Central Questions of Computer Science

- Which problems can be solved by algorithmic processes?
- How can algorithm discovery be made easier?
- How can techniques of representing and communicating algorithms be improved?
- How can characteristics of different algorithms be analyzed and compared?

Central Questions of Computer Science
(continued)

- How can algorithms be used to manipulate information?
- How can algorithms be applied to produce intelligent behavior?
- How does the application of algorithms affect society?

The central role of algorithms in computer science



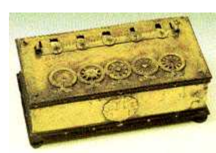
Abstraction

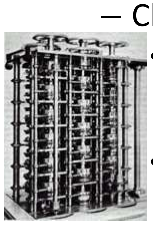
- **Abstraction:** The distinction between the external properties of an entity and the details of the entity's internal composition
- **Abstract tool:** A "component" that can be used without concern for the component's internal properties

- **Abstraction simplifies many aspects of computing and makes it possible to build complex systems**

A Brief History of Computing

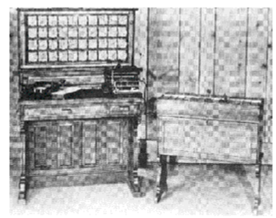
- Roots in Mathematical Sciences and computational devices
  - Abacus, counting device, state
  - Blaise Pascal, the Pascaline 1642
    - Manual gear system to add numbers
  - Charles Babbage
    - Difference Engine designed in 1812
      - Could not be built using the tools of the era
      - Eventually built later using modern tools
    - Analytic Engine 1823, steam-powered more general computational device with conditional controls
      - Also too complex to build in the 19th century





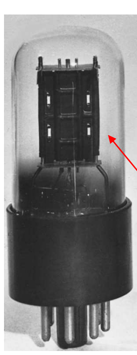
Roots of Computing…

- Herman Hollerith's Tabulating Machine
  – Former MIT lecturer, developed a machine to read punch cards
  – Inspired by a train conductor to punch tickets
  – Used in the 1890 US Census
  – Company became IBM in 1924

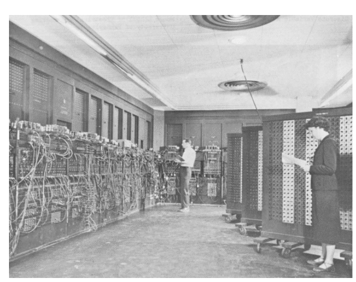


Roots of Computing…

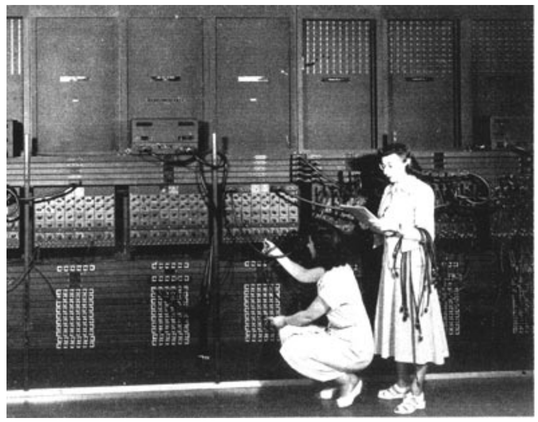
- 1940, Conrad Zuse's Z3
  – First computing machine to use binary code, precursor to modern digital computers
- 1944, Harvard Mark I, Howard Aiken
- 1946, ENIAC, first all digital computer
  – Ushered in the "Mainframe" era of computing
  – "First Generation"
  – 18,000 vacuum tubes



Similar to a lightbulb but plate in middle controls flow of electrical current



- **On the ENIAC, all programming was done at the digital logic level.**
- **Programming the computer involved moving plugs and wires.**

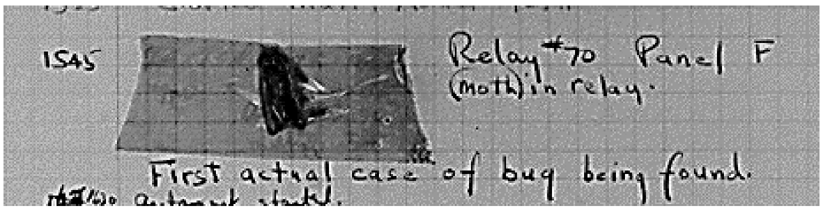


Roots of Computing…

- 1945: John von Neumann defines his architecture for an “automatic computing system”
  - Basis for architecture of modern computing
    - Computer accepts input
    - Processes data using a CPU
    - Stores data in memory
      - Stored program technique, storing instructions with data in memory
    - Produces output
- Led to the EDVAC and UNIVAC computers

Roots of Computing…

1951, UNIVAC, Universal Automatic Computer



When we say there is a "bug" in the program, we mean it doesn't work right… the term originated from an actual moth found in the UNIVAC by Grace Hopper

The Second Generation: Transistors

- Invented 1947, Bell Labs: Bardeen, Shockley, Brattain
- 1958 -1964
- Transistors generate less heat
- Transistors are smaller, faster, and more reliable
- First transistors smaller than a dime
- UNIVAC II built using transistors

The Third Generation: Integrated Circuits (IC)

- 1964 -1990
- Multiple transistors on a single chip
- IBM 360 - First mainframe to use IC
- DEC PDP-11 - First minicomputer
- End of mainframe era, on to the minicomputer era

Integrated Circuit

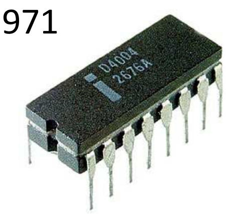
- Invented at TI by Jack Kilby, Bob Noyce
- "*What we didn't realize then was that the integrated circuit would reduce the cost of electronic functions by a factor of a million to one, nothing had ever done that for anything before*" - Jack Kilby

Minicomputer Era

- Made possible by DEC and Data General Corporation, IBM
- Medium-sized computer, e.g. DEC-PDP
- Much less expensive than mainframes, computing more accessible to smaller organizations
- Used transistors with integrated circuits

Personal Computer Era

- First microprocessor, Intel 4004 in 1971
- MITS Altair “kit” in 1975
- Apple in 1976
- IBM PC in 1981 using 8086
- Macintosh in 1984, introduced the GUI (Graphical User Interface) we still use today
  - Some critics: Don Norman on complexity
  - Next interface delegation instead of direct manipulation?



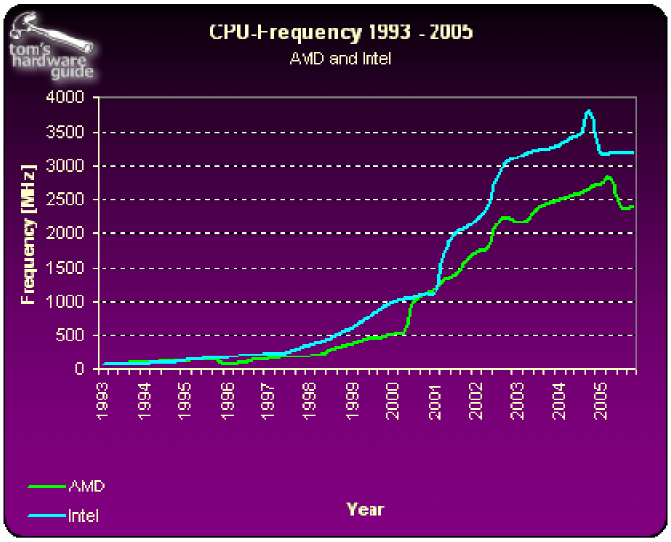
Today: Internetworking Era?

- Computer as communication device across networks
- World Wide Web, Internet
- Publishing, data sharing, real-time communications

Supercomputers

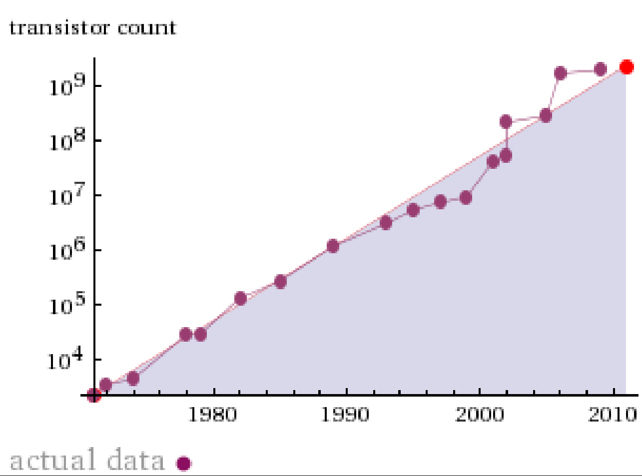
- The most powerful and expensive computers
- Contain numerous very fast processors that work in parallel
  - IBM Roadrunner
    - 1,105 TeraFlops (Floating Point Operations/Second)
    - 12,960 IBM PowerXCell 8i and 6,480 AMD Opteron dual-core processors
  - At 2 TeraFlops, could do in 1 second what would take every man, woman, and child 125 years work nonstop on hand calculators
- Used by researchers and scientists to solve very complex problems
- Cost millions of dollars

CPU Clock Speeds



Moore's Law

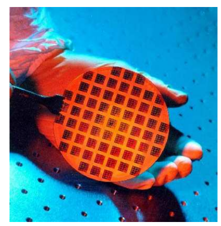
1965: Computing power doubles ~ every 18 months



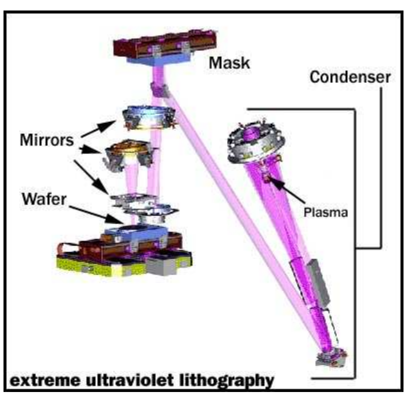
Chip Production

- Ingot of purified silicon – 1 meter long, sliced into thin wafers
- Chips are etched – much like photography
  - UV light through multiple masks
  - Circuits laid down through mask
- Process takes about 3 months

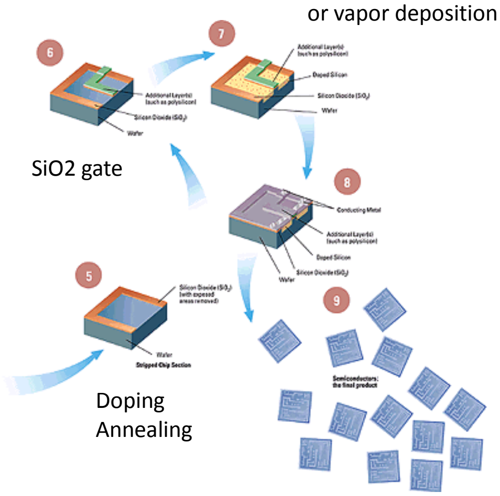
View of
Cross-Section



Fabrication

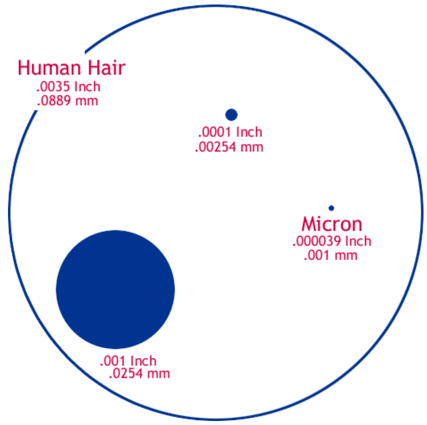


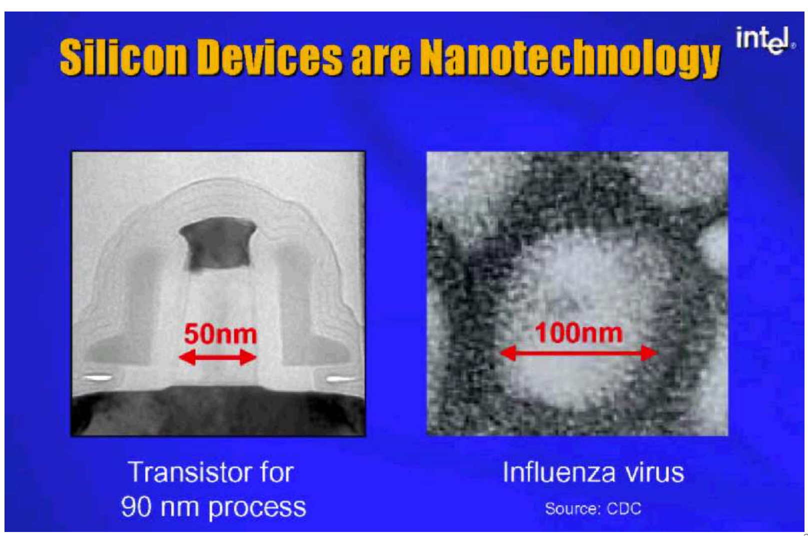
"wires" – chemical or vapor deposition



The Shrinking Chip

- Human Hair: 100 microns wide
  - 1 micron is 1 millionth of a meter
- Bacterium: 5 microns
- Virus: 0.8 microns
- Early microprocessors: 10-15 micron technology
- 1997: 0.35 Micron
- 1998: 0.25 Micron
- 1999: 0.18 Micron
- 2001: 0.13 Micron
- 2003: 0.09 Micron
- 2007: 0.065 Micron
- 2009: 0.045 Micron
- Physical limits believed to be around 0.016 Microns, should reach it around 2018





30