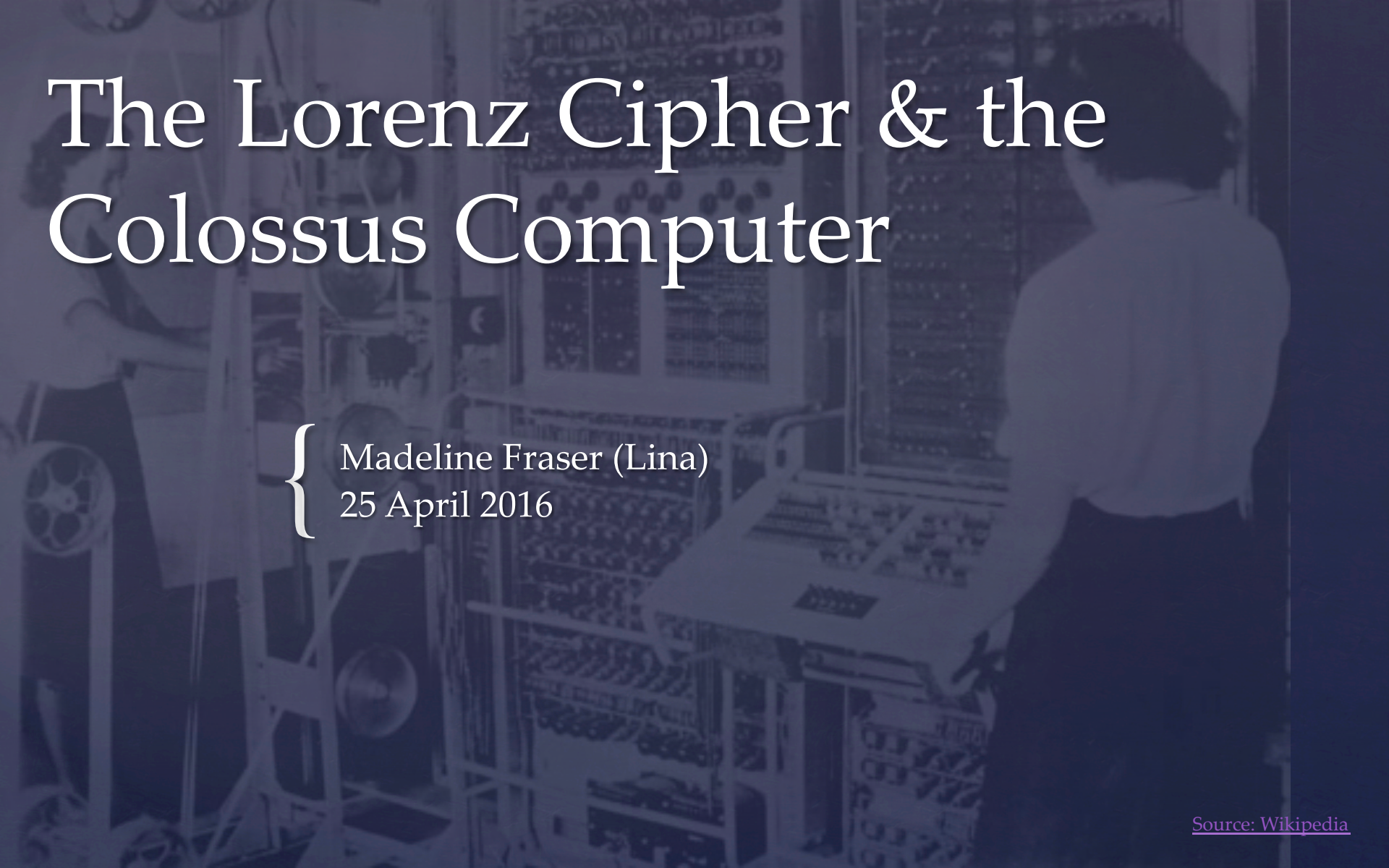
The Lorenz Cipher & the Colossus Computer

Madeline Fraser (Lina)
25 April 2016



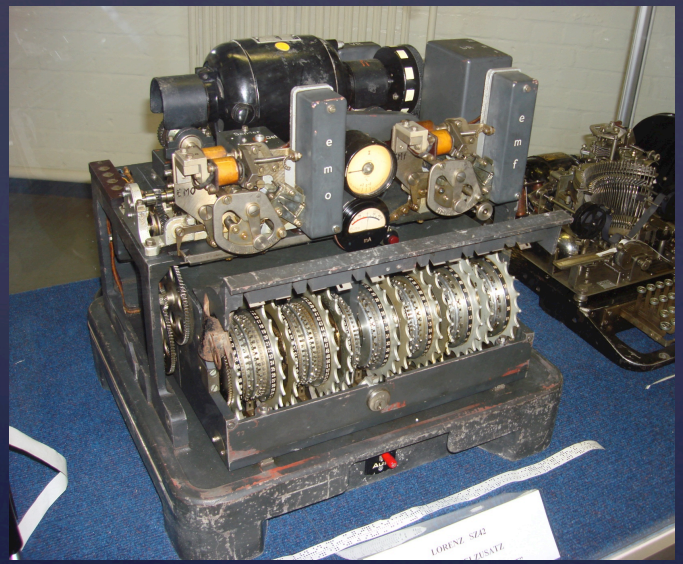
Source: Wikipedia

German Encryption: The Lorenz Cipher

The 32-Character Baudot Code
represented in binary using crosses (1) and dots (0)

	0	1					2									3										4					5
	/	E	4	9	3	T	A	S	D	Z	I	R	L	N	H	O	U	J	W	F	Y	B	C	P	G	M	K	Q	+	X	V	8
1	•	×	•	•	•	•	×	×	×	×	•	•	•	•	•	•	×	×	×	×	×	×	•	•	•	•	×	×	×	×	•	×
2	•	•	×	•	•	•	×	•	•	•	×	×	×	•	•	•	×	×	×	•	•	•	×	×	×	•	×	×	×	•	×	×
3	•	•	•	×	•	•	•	×	•	•	×	•	•	×	×	•	×	•	•	×	×	•	×	×	•	×	×	×	•	×	×	×
4	•	•	•	•	×	•	•	•	×	•	•	×	•	×	•	×	•	×	•	×	•	×	×	•	×	×	×	•	×	×	×	×
5	•	•	•	•	•	×	•	•	•	×	•	•	×	•	×	×	•	•	×	•	×	×	•	×	×	×	•	×	×	×	×	×
ª	#	3	#	#	#	5	-	'	#	+	8	4	)	,	*	9	7	#	2	*	6	?	:	0	*	.	(	1	#	/	=	#

Source: Rutherford Journal



Source: Wikipedia

The Lorenz Cipher Attachment
&
the Lorenz Teleprinter

Source: Flickr

The Encryption and Decryption Process

Lorenz encryption

S: 10100 M: 00111 I: 01100 T: 00001 H: 00101 ← Plaintext

\+

J: 11010 K: 11110 A: 11000 V: 01111 L: 01001 ← Key stream

↓

C: 01110 W: 11001 S: 10100 C: 01110 I:01100 ← Ciphertext (encrypted)

Result: SMITH + JKAVL → CWSCI

Lorenz decryption

C: 01110 W: 11001 S: 10100 C: 01110 I:01100 ← Ciphertext (encrypted)

\+

J: 11010 K: 11110 A: 11000 V: 01111 L: 01001 ← Key stream

↓

S: 10100 M: 00111 I: 01100 T: 00001 H: 00101 ← Plaintext

Result: CWSCI + JKAVL → SMITH

Baudot Code

Binary	Letter
00000	Blank
00001	T
00010	CR
00011	O
00100	Space
00101	H
00110	N
00111	M
01000	Line Feed
01001	L
01010	R
01011	G
01100	I
01101	P
01110	C
01111	V
10000	E
10001	Z
10010	D
10011	B
10100	S
10101	Y
10110	F
10111	X
11000	A
11001	W
11010	J
11011	Figure Shift
11100	U
11101	Q
11110	K
11111	Letter Shift

XOR Truth Table

Inputs		Outputs
X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	0

Example idea credit: Codes and Ciphers by Anthony Sale

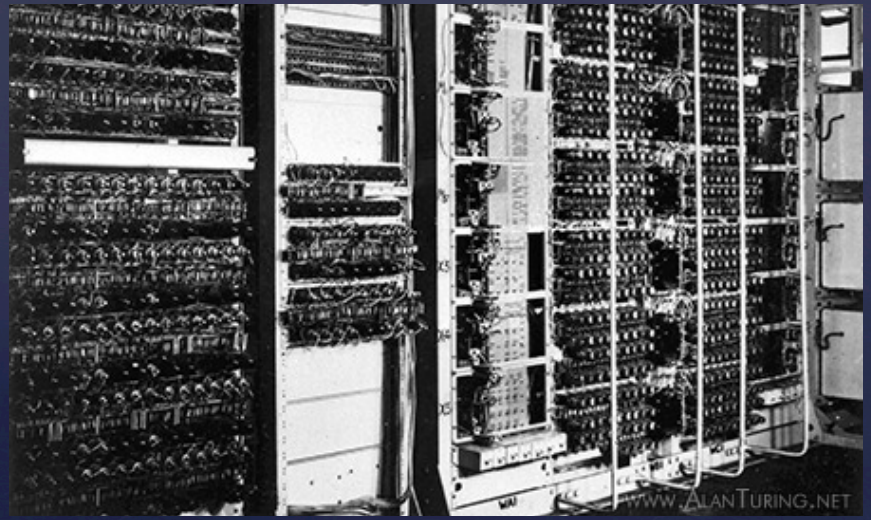
Truth table: ElectronicsHub Key: Stanford CS

The Colossus Computer

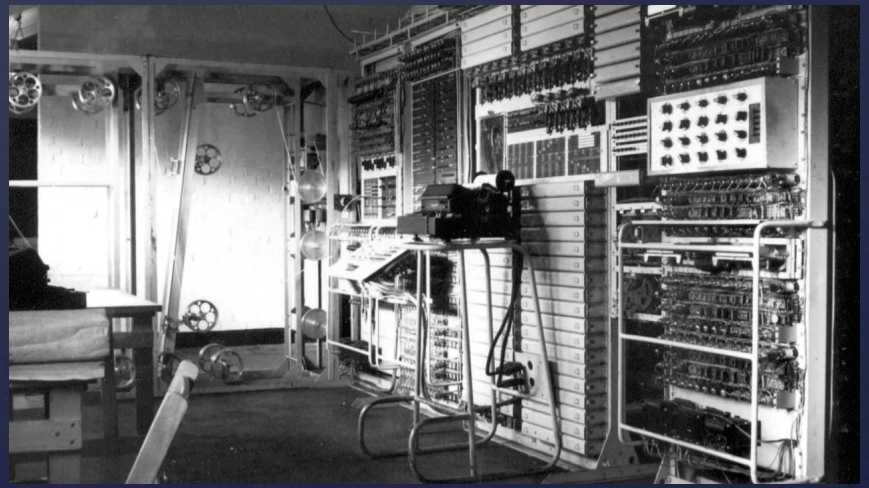
- Made with 2,500 vacuum tubes
- Wheel positions simulated using thyratron rings
- Programmable using plug panels and switches
- Processing speed: 5,000 characters/second

Source: Wikipedia

Tommy Flowers



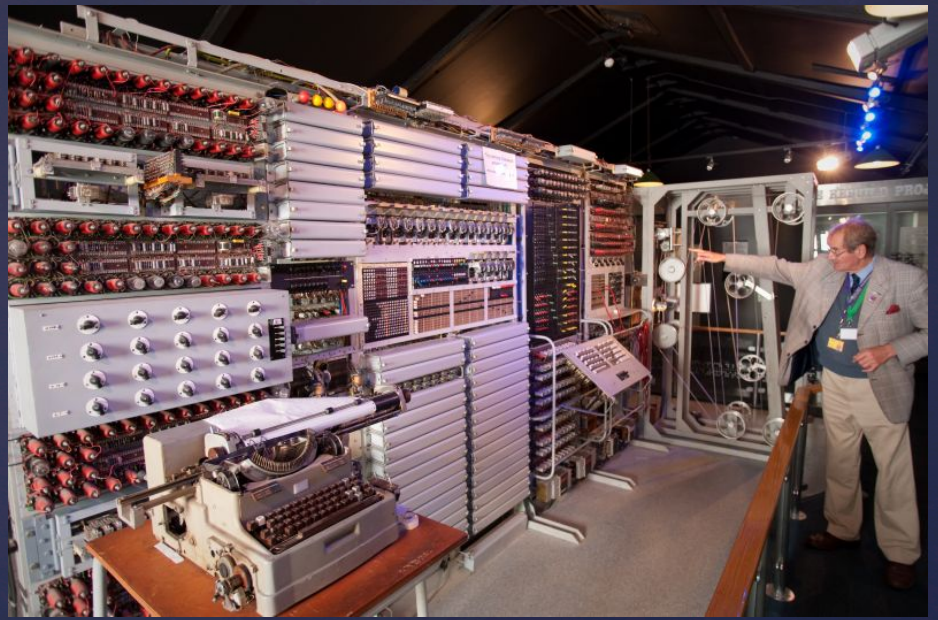
Source: colossuscomputer.com



Source: YouTube

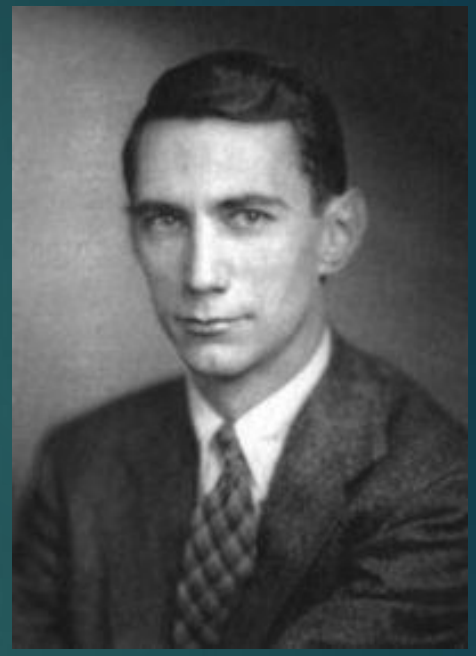
Conclusion

- Shortened WWII by approximately 2 years
- Classified and destroyed
- Rebuild finished in 2008
- Goal for the write-up: To investigate & understand the internal mechanisms of Colossus in more depth



Source: The National Museum of Computing

Claude Shannon's Information Theory



BY JI WON CHUNG

APRIL 25, 2015

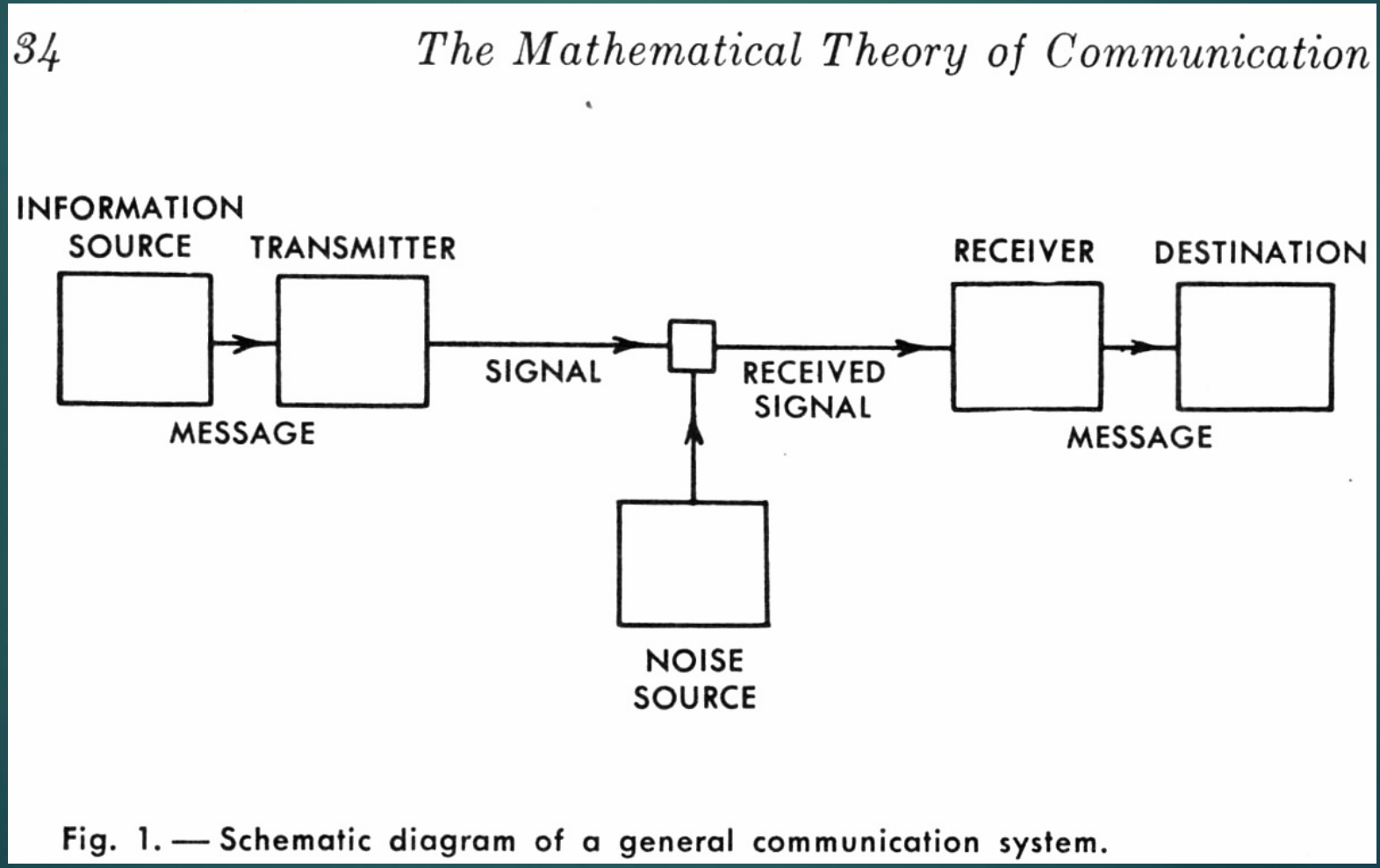
Significance in Context

- Nyquist 1924 & Hartley 1928
- What is information?
- Unifying Concept
- Simple

Terms

- **Information**: #bits/symbol
  - What CAN you send, not what do you send
- **Entropy**: quantitative measure of information uncertainty
- **Communication**: transmission of info across space and time
- **Channel Capacity:** how much info can be sent
- **Source Coding Theorem:** number of bits needed to send the message without much distortion
- **Channel Coding Theorem**: error reduced if info rate is < channel capacity

A Mathematical Theory of Communication (1948)



Impact

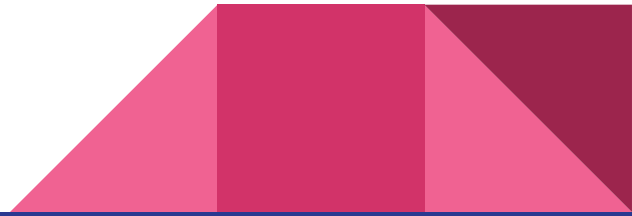
- Coding theory
- Issues of Transactions on Information Theory
- 1970s revival
- Determined digital communication:
  - Data compression
  - Data encryption
  - Data correction

An Introduction to C Programming

CSC 103
Hannah Kwon
April 25, 2016

A BRIEF HISTORY

- Was formulated in early 1970s by an American computer scientist, Dennis Ritchie, who worked at Bell Labs (AT&T)
- Ritchie started off trying to make new file system → an intricate system called UNIX, all written through assembly language
- Its “predecessor” → B (devised in 1969~1970 by another computer scientist named Ken Thompson)
- B had its pros → was efficient and was an upgrade from assembly language
- However, it also had its cons: Thus, B → C

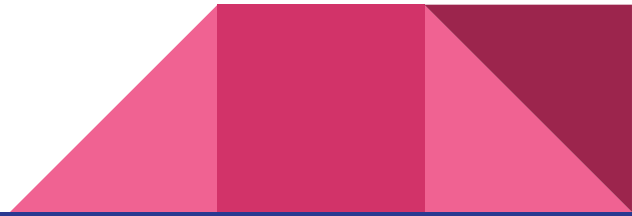


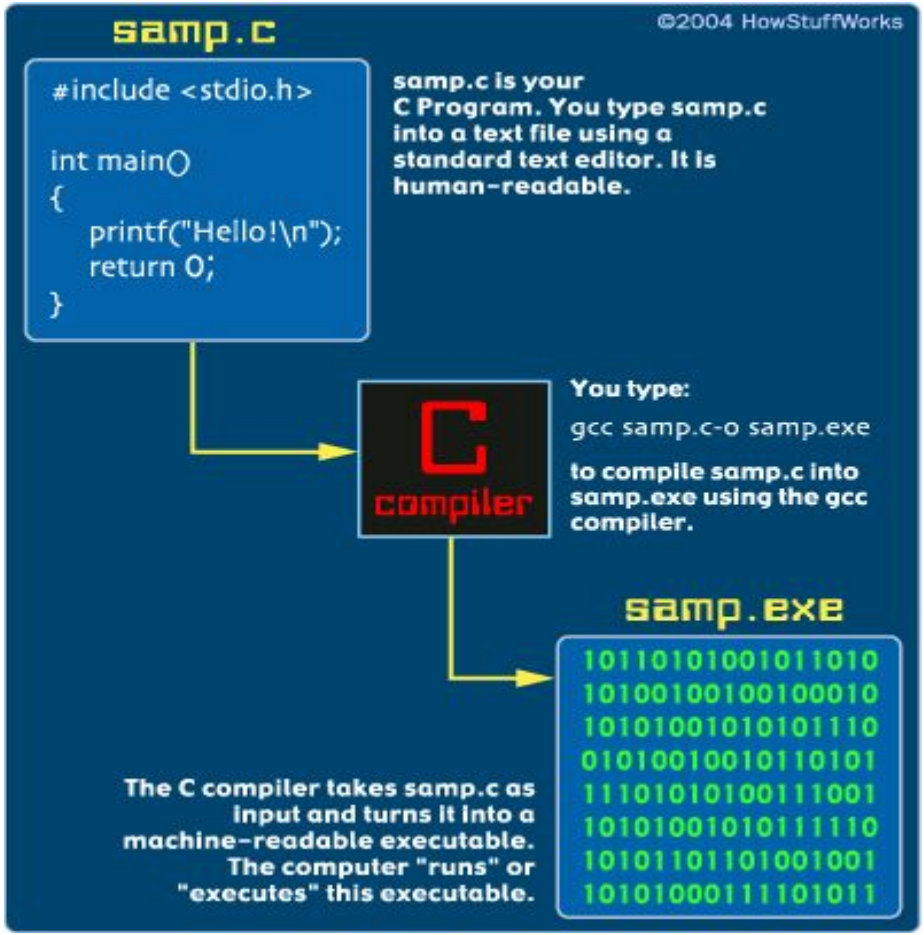
HOW DOES C WORK?

- C is a “compiled language”
- After writing the program, it gets run through a C compiler → changes programs into “executable”
- C itself: “human-readable” form
- “Executable” : “machine-readable” form

OTHER IMPORTANT USES:

- Operating systems
- Databases
- Interpreting language





EXAMPLE OF A C PROGRAMMING CODE

- “Include” Line: standard in & standard out
- Int: Integers
- “Main” Line
- Printf- Output
- Return 0- Make sure no error

CONCLUSION & FUTURE RESEARCH

- Why is C programming so important?
  - ★ Crucial advance in the field of computer science
  - ★ composes of the most basic building blocks
  - ★ leads the way for C++ and future programming languages
- Paper Topic
  - ★ Potential future prospects of C language
  - ★ More in-depth history- how programming has evolved over time
  - ★ Explain how to interpret harder C programming codes

REFERENCES

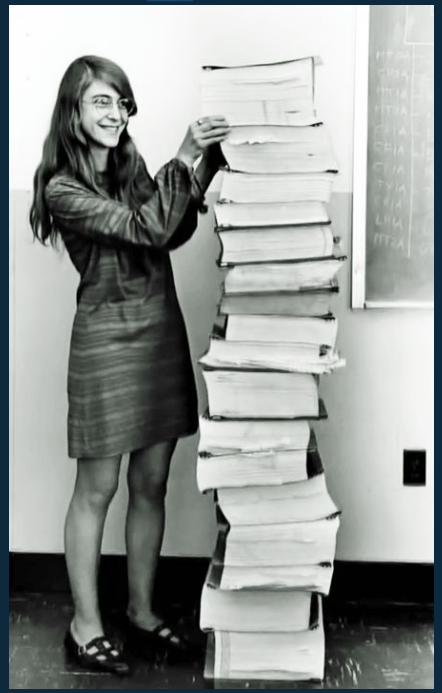
- http://computer.howstuffworks.com/c.htm
- https://www.bell-labs.com/usr/dmr/www/bintro.html
- https://www.le.ac.uk/users/rjm1/cotter/page_05.htm
- https://www.codingunit.com/the-history-of-the-c-language
- https://lh3.googleusercontent.com/3gI9l3yQynt2cj1MFdTZbaYE0VK056s-lvE4iejCCZQ1_-S8v3ZGDCPsIhtQsOB8Kb8i=w300
- http://s.hswstatic.com/gif/c-compile.gif

Pioneering Women in Computer Science

Tasha Binkowski

4/25/16

Narrowing Focus



ADA LOVELACE

1815-1852

First Conceptual Programmer

GRACE HOPPER

1906 - 1992

Higher-level Programming Languages

RADIA PERLMAN

Born 1951

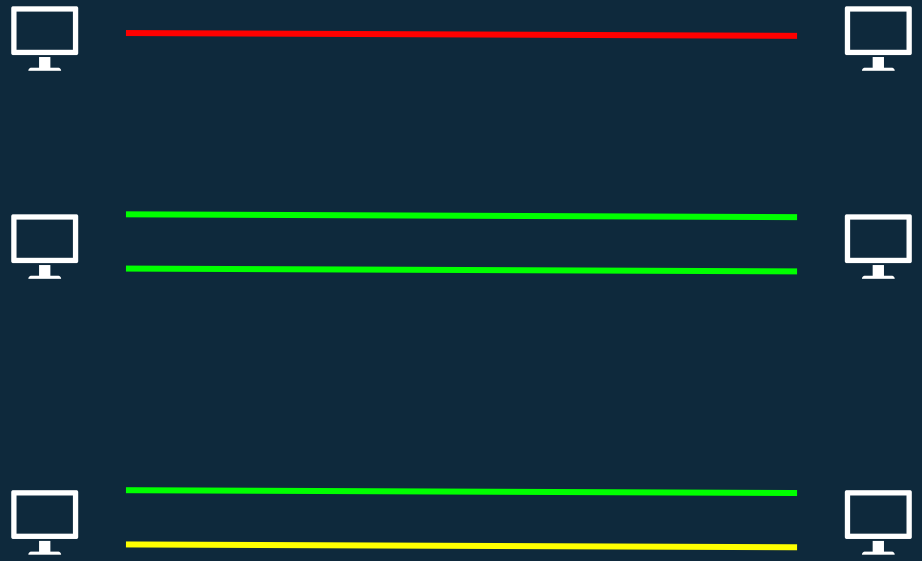
Developed Algorithm behind STP

What is it?

➔ Spanning Tree Protocol :

- Layer 2 (Data Link) protocol where bridges are used to interconnect multiple LANs (WAN) or parts of one LAN
- passes data back and forth to find out how the switches are organized on the network
- takes all the information it gathers and uses it to create a logical tree
  - the bridges exchange information so that only one of them will handle a given message that is being sent between two computers within the network
  - prevents the condition known as a bridge loop

Spanning Tree Protocol Example



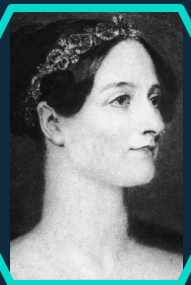
Preventing Loops & Providing Path Redundancy by

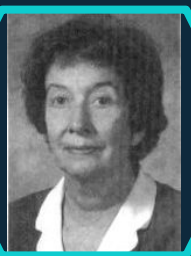
Creating a Tree and

Only Allowing One Active Path at a Time

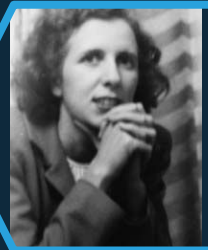
Conclusion

There really are too many to list



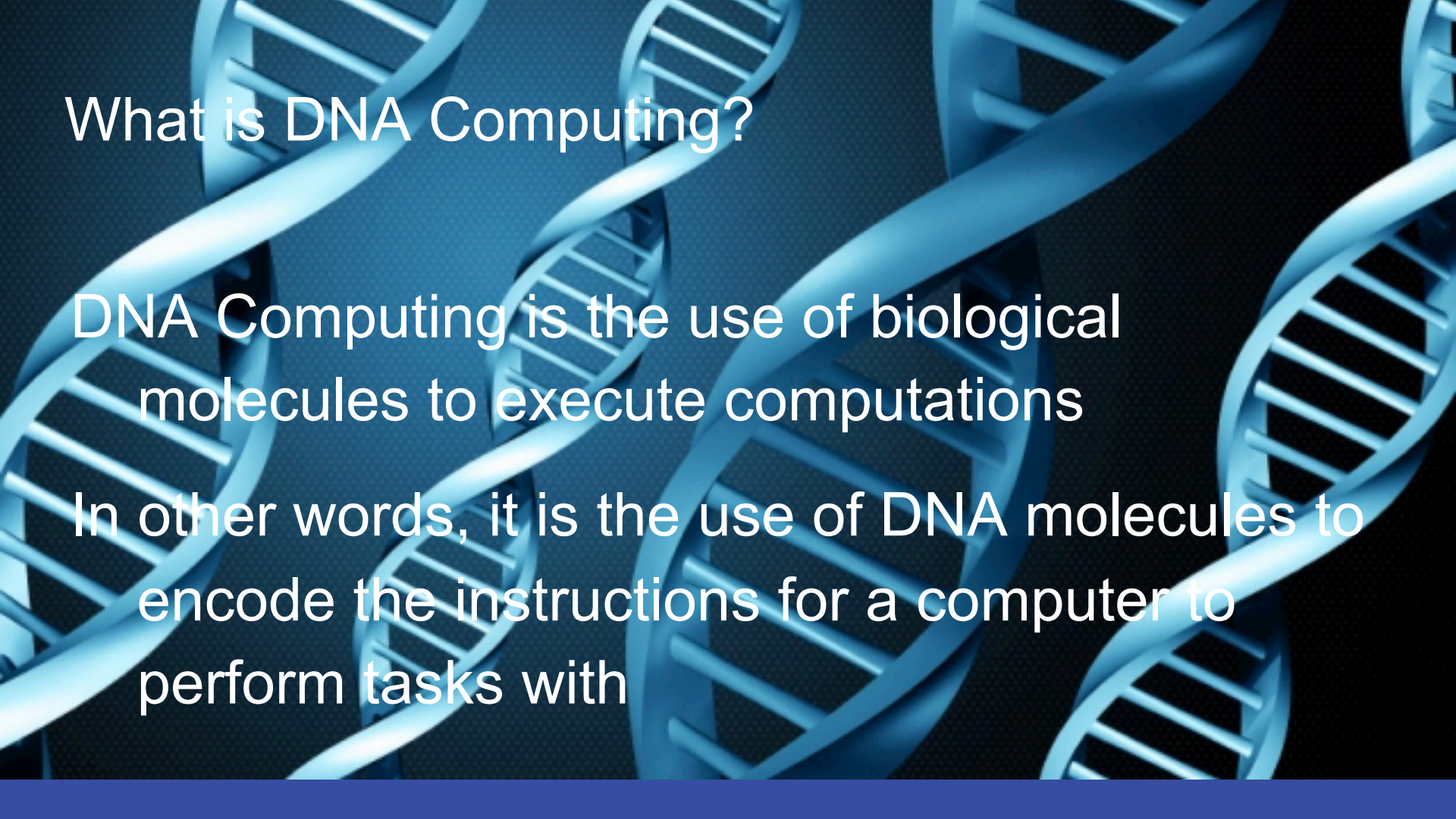





DNA Computing

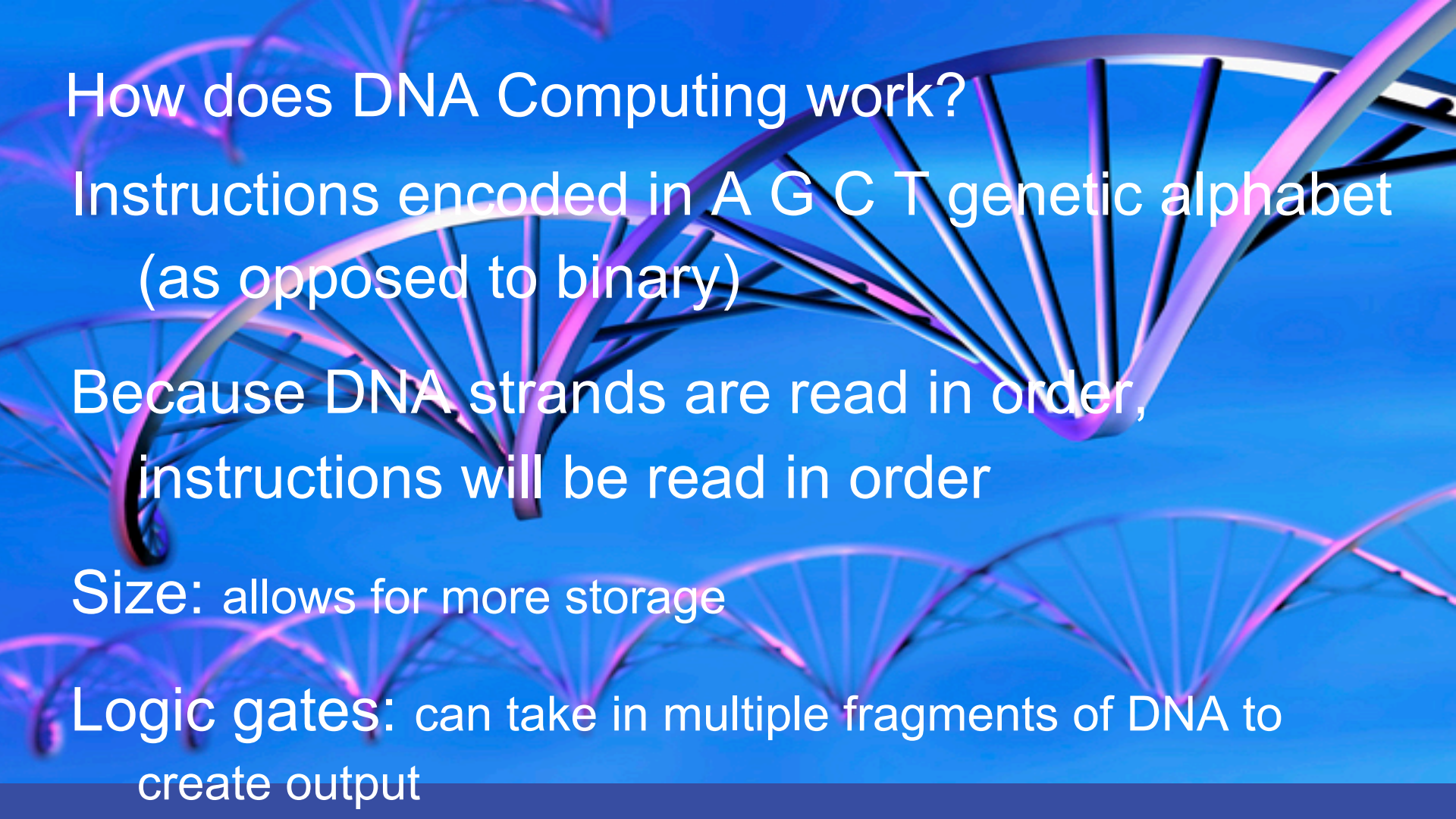
Raphaela Tayvah
25 April 2016



What is DNA Computing?

DNA Computing is the use of biological molecules to execute computations

In other words, it is the use of DNA molecules to encode the instructions for a computer to perform tasks with



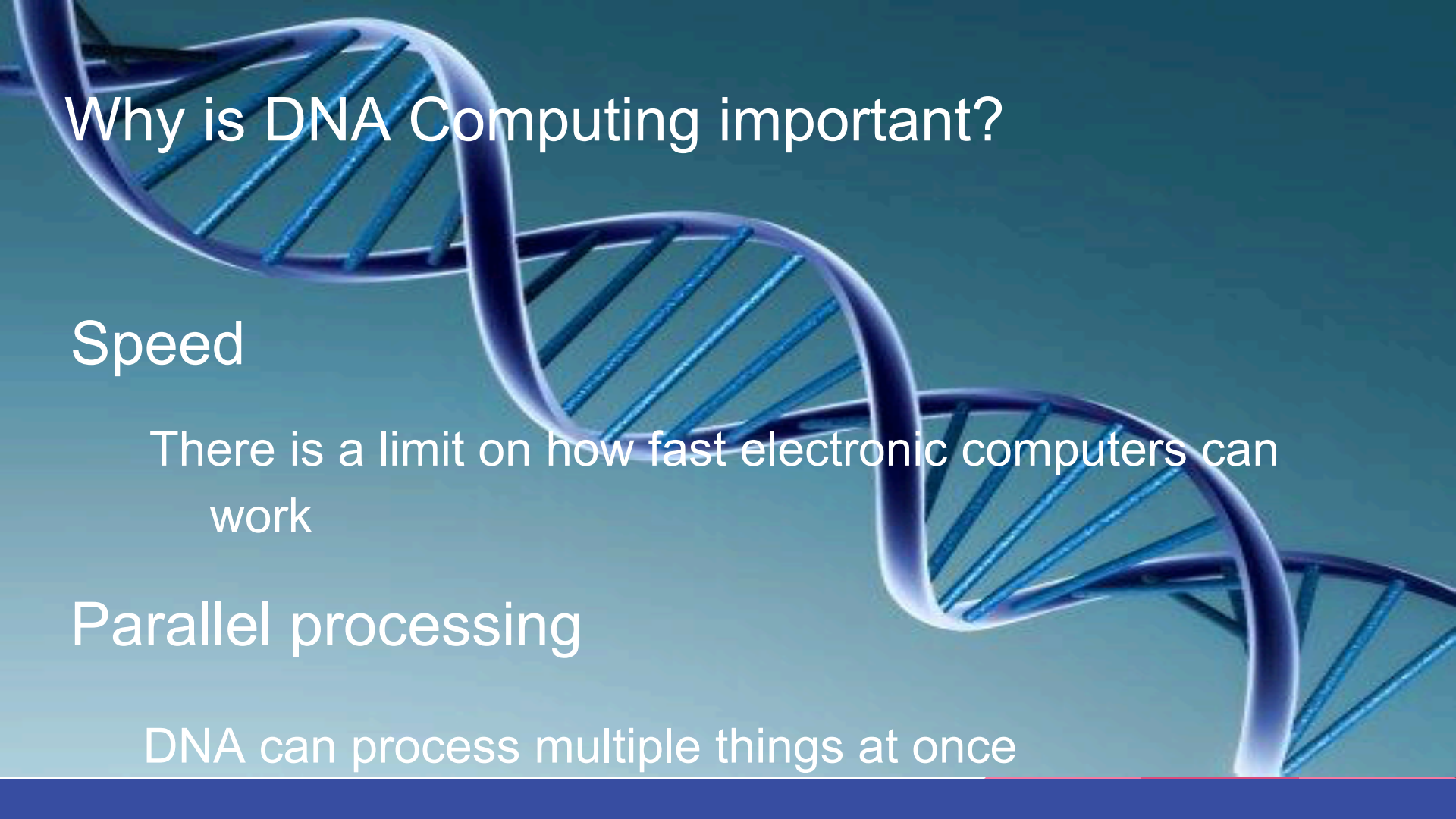
How does DNA Computing work?

Instructions encoded in A G C T genetic alphabet (as opposed to binary)

Because DNA strands are read in order, instructions will be read in order

Size: allows for more storage

Logic gates: can take in multiple fragments of DNA to create output



Why is DNA Computing important?

Speed

There is a limit on how fast electronic computers can work

Parallel processing

DNA can process multiple things at once

Sources

Gheorghe Paun, Grzegorz Rozenberg, Arto Salomaa, *DNA Computing: New Paradigms* (Springer Science & Business Media, 2005).

Fumiaki Tanaka, Masashi Nakatsugawa, Masahito Yamamoto,Toshikazu Shiba, Azuma Ohuchi, Developing Support System for Sequence Design in DNA Computing in *DNA Computing* (Springer Berlin Heidelberg, 2002) pp 129-137.

Gheorghe Păun, “Computing with Membranes” in *Journal of Computer and System Sciences* *Volume 61, Issue 1* (Bucharest, Romania 2000) pp 108-143.

http://www.exploredna.co.uk/dna-computing-benefits.html

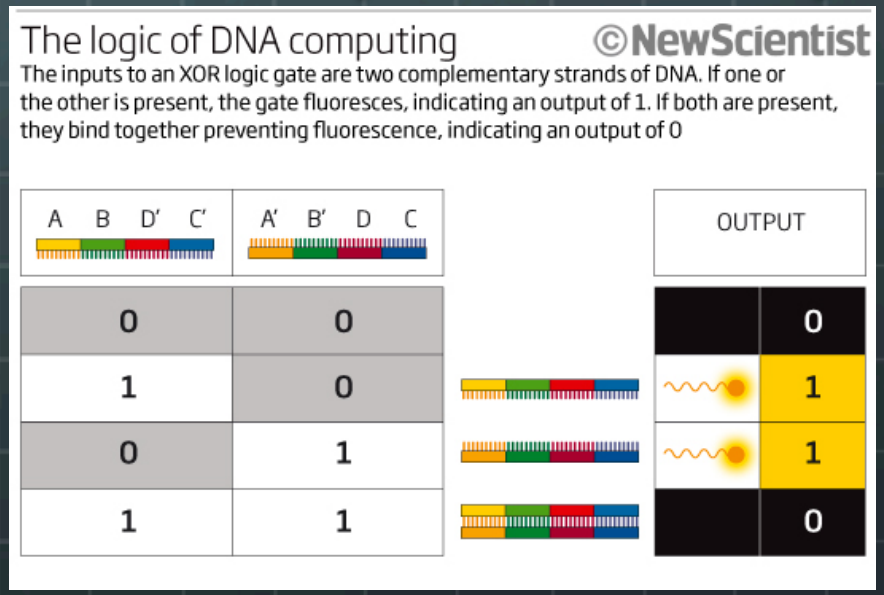
http://news.discovery.com/tech/future-computers-may-be-dna-based-110521.htm

http://computer.howstuffworks.com/dna-computer.htm

http://www.britannica.com/technology/DNA-computing

DNA Computing:

- Invented by Leonard Adleman in 1994 at the University of Southern California
- Combines DNA, biochemistry, and molecular biology hardware to solve complex problems (the first one solved was the seven point Hamilton Path Problem) and comes away with multiple solutions
- Connected with Turing machines

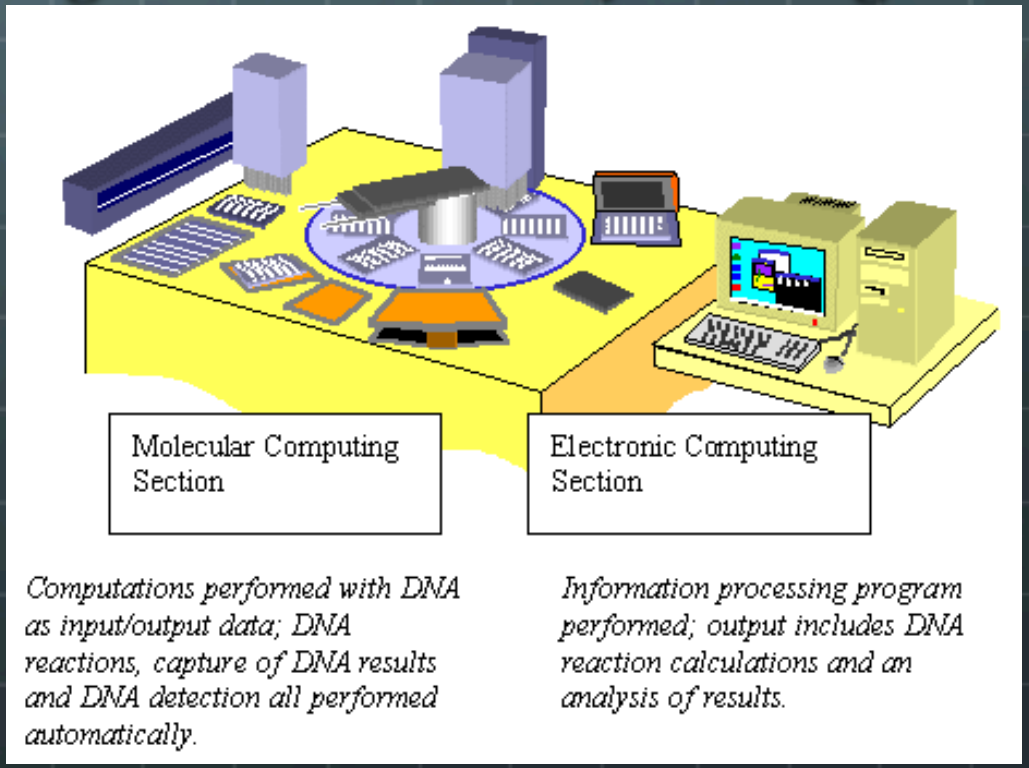


The Technicalities:

- DNA computing works because DNA has its own coding mechanism (4-key components of the DNA molecule, storing genetic "codes")
- Enzymes react with strands of DNA and cause chain chemical reactions
- Theoretically, computers using DNA computing would have much faster data transfer speeds and hold more memory
- Right now, DNA computing can take hours or days, but it can make "a high amount of multiple parallel computations" (different possible solutions are created simultaneously)

Conclusion:

- Since Adleman's first proposal, many developments in the field have been made
- Scientists are still researching more efficient ways of using the technology
- Still a long way to go before it is on the market



CSC103 Betty Cui

Sorting Algorithms

What Is Sorting?

- Sorting is ordering a list of objects so that they are organized in desired ways efficiently.

- Internal sorting
  - takes place in the main memory, where we can take advantage of the random access nature of the main memory
- External sorting
  - is necessary when the number and size of objects are prohibitive to be accommodated in the main memory.

Internal Sorting

- **Bubble Sort**
- **Insertion Sort**
- **Selection Sort**
- Shell Sort
- Quick Sort
- Heap Sort

External Sorting

- **Mergesort**
- Radix Sort
- Polyphase Sort

Comparison of Efficiency

Legend

Excellent | Good | Fair | Bad | Horrible

Array Sorting Algorithms

Algorithm	Time Complexity		
	Best	Average	Worst
Quicksort	O(n log(n))	O(n log(n))	O(n^2)
Mergesort	O(n log(n))	O(n log(n))	O(n log(n))
Timsort	O(n)	O(n log(n))	O(n log(n))
Heapsort	O(n log(n))	O(n log(n))	O(n log(n))
Bubble Sort	O(n)	O(n^2)	O(n^2)
Insertion Sort	O(n)	O(n^2)	O(n^2)
Selection Sort	O(n^2)	O(n^2)	O(n^2)
Shell Sort	O(n)	O((nlog(n))^2)	O((nlog(n))^2)
Bucket Sort	O(n+k)	O(n+k)	O(n^2)
Radix Sort	O(nk)	O(nk)	O(nk)

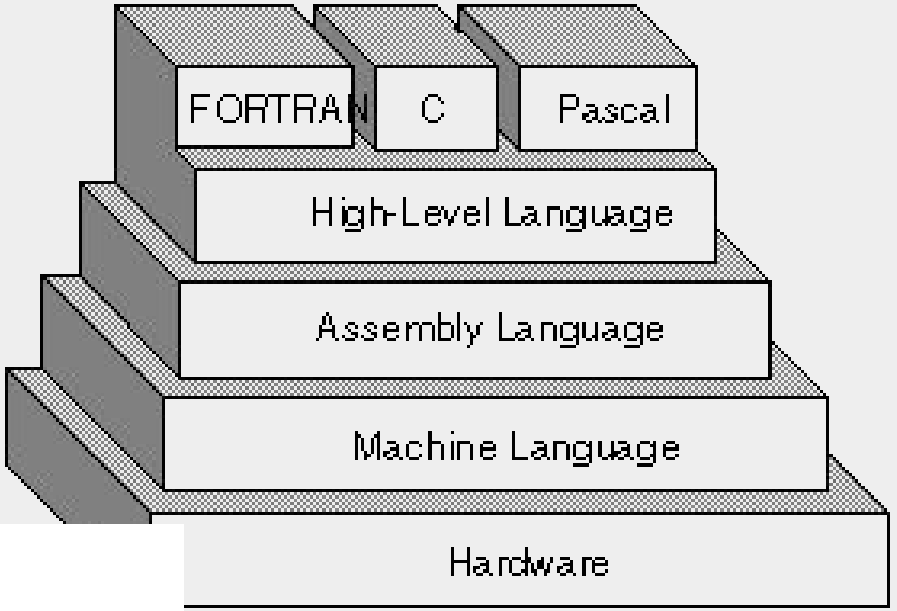
Conclusion

- Investigate more about different types of sorting
- & more about the comparison of efficiency
- How do they react differently to different situations
- Efficiency: not only time but also space complexity

Assembly Language

By Dardalie Brooks
CSC103_Spring 2016

- Assembly is a “low-level” machine language.
- Machine code details instructions carried out by the CPU (processor).



Assembly Language	Machine Code
SUB AX,BX	001010111000011
MOV CX,AX	100010111001000
MOV DX,0	10111010000000000000000

Addressing Modes

Example declarations:

```
.DATA
var   DB 64        ; Declare a byte, referred to as location var, containing the value 64.
var2  DB ?         ; Declare an uninitialized byte, referred to as location var2.
      DB 10        ; Declare a byte with no label, containing the value 10. Its location is var2 + 1.
X     DW ?         ; Declare a 2-byte uninitialized value, referred to as location X.
Y     DD 30000     ; Declare a 4-byte value, referred to as location Y, initialized to 30000.
```

3.

- Directives (DB, DW, DD) are used to declare static data regions. They declare one, two, and four byte data locations, respectively
- **static variable** is a variable whose "lifetime" extends across the entire run of the program.

Assembly Instructions

Adding a series

WHEN X= 1 and Y =5

```
MOV A, [x]     copy value of x into A
MOV B, [y]     Copy value of y into B
.loop:
   ADD A,B     This line says “Add B to A”
   ADD B,1     This line says “Add 1 to B”
  CMP A, 15    This line says “compare A to 15; Sets zero (Z) flay to 1 (true) when  A= 15
   JNZ  .loop  This line tells program to stop when Z flay = 1 ( above is true)
   x: DB 0
   y: DB 1          DB’s tell us where x and y start
```

Assembly in my near future

- Pros and Cons of Assembly
- Assembly beyond an introductory comp. sci. class?
- Assembly code → assembler → machine code → CPU

Citations:

1. “Assembly Language,” Webopedia, accessed April 23,2016, http://www.webopedia.com/TERM/A/assembly_language.html

2. Cutajar,John, “Intermediate 8086 Assembly Language programming,” (Slideshow presented as part of a class at the University of Malta junior College, March 17, 2012).
3. Evans, David, “Guide to x86 Assembly,” University of Virginia, September 23, 2015.
4. Hyde, Randall. *The Art of Assembly Language; 2nd Edition*. No Starch Press, 2010.

SURGICAL ROBOTS

ALEXIS COHEN

APRIL 25TH, 2016

BACKGROUND

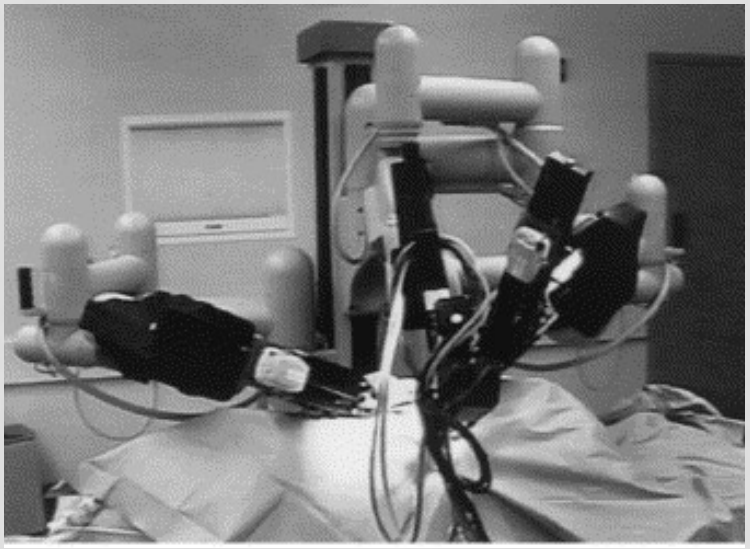
Classification of Robots:
1. Supervisory
2. Telesurgical
3. Shared-control

Disadvantages:
1. Expenses
2. No tactile feedback
3. Less flexibility with positioning

Advantages:
1. Magnification
2. No tremors
3. 3-D Vision
4. Larger range of motion
5. Ergonomically better for surgeons

THE DA VINCI SYSTEM

Hardware:
1. Surgical Cart
2. Vision Cart
3. Surgeon Console



How is it Used?
-Minimally Invasive Surgeries
-Training for Techniques
-Many types of surgeries

THE DA VINCI SYSTEM

How it Works:

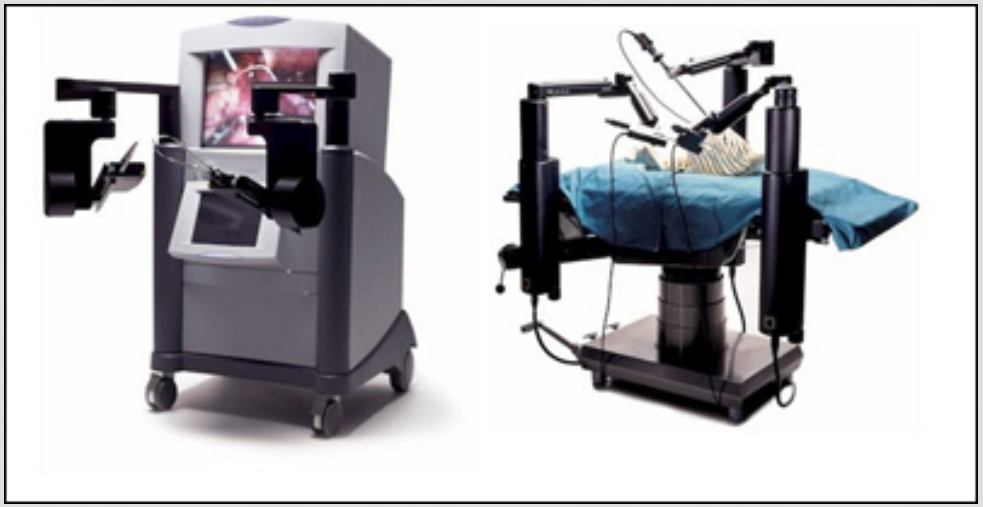
-EndoWrist Instruments:
- -Range of Freedom
- -Motion and Flexibility
- -Small Incisions vs. Open Surgery

-Vision:
- -Uses a two channel endoscope
- -Cannot see the rest of the operating room

Technical Difficulties:
-Recoverable vs. Non-recoverable Errors

WHAT'S NEXT?

I am hoping to look into a specific surgery that surgical robots are used in, such as pancreatic surgery.



There are also different robotic systems used in surgeries that I want to investigate, such as ZEUS.

CITATIONS

Ady, Justin, and Vincent P. Laudone. *Introduction to Robotic Surgery*. Springer International, 2015. Print.

Bruns, Nicholas E., Oliver S. Soldes, and Todd A. Ponsky. "Robotic Surgery May Not "Make the Cut" in Pediatrics." *Frontiers in Pediatrics*. Frontiers Media S.A., 12 Feb. 2015.

Gyung Tak Sung, Inderbir S Gill, Robotic laparoscopic surgery: a comparison of the da Vinci and Zeus systems, Urology, Volume 58, Issue 6, December 2001, Pages 893-898

Kroh, Matthew, and Sricharan Chalikonda. *Essentials of Robotic Surgery*. Springer International, 2014. Print.

Ross, Howard M., Sang W. Lee, Bradley J. Champagne, Alessio Pigazzi, and David E. Rivadeneira. *Robotic Approaches to Colorectal Surgery*. Springer International, 2015. Print.

Endowrist Picture from: www.birminghambowelclinic.co.uk

ZEUS Picture from: www.prweb.com

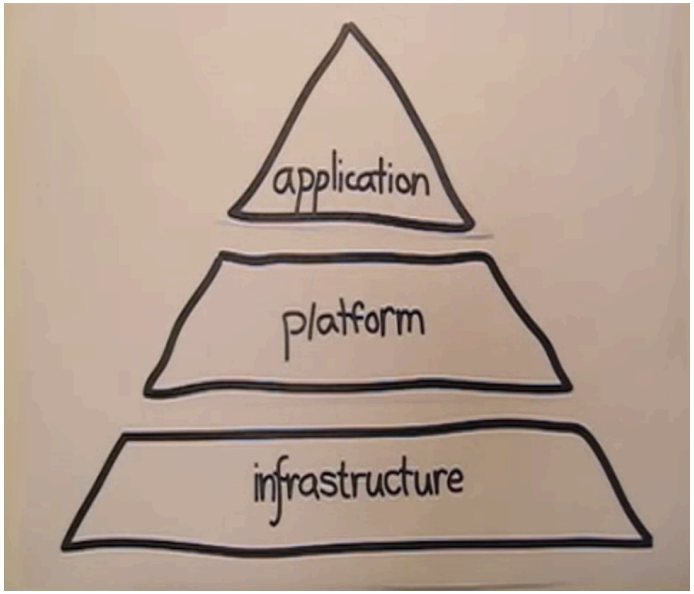
Cloud Computing

Echo Zhang

04/25/2016

CSC 103

What is Cloud Computing and how does it work?



Why is it popular?

- Scalability
- Instant
- Save Money

Gmail™ by Google™

NETFLIX

Disadvantages of Cloud Computing

- Privacy
- Security
- Control
- Internet Access

Conclusion…

The History of Statistical Computation

JULIANNA CALABRESE

APRIL 25TH, 2016

"The utmost confusion is caused when people argue on different statistical data."
–Winston Churchill

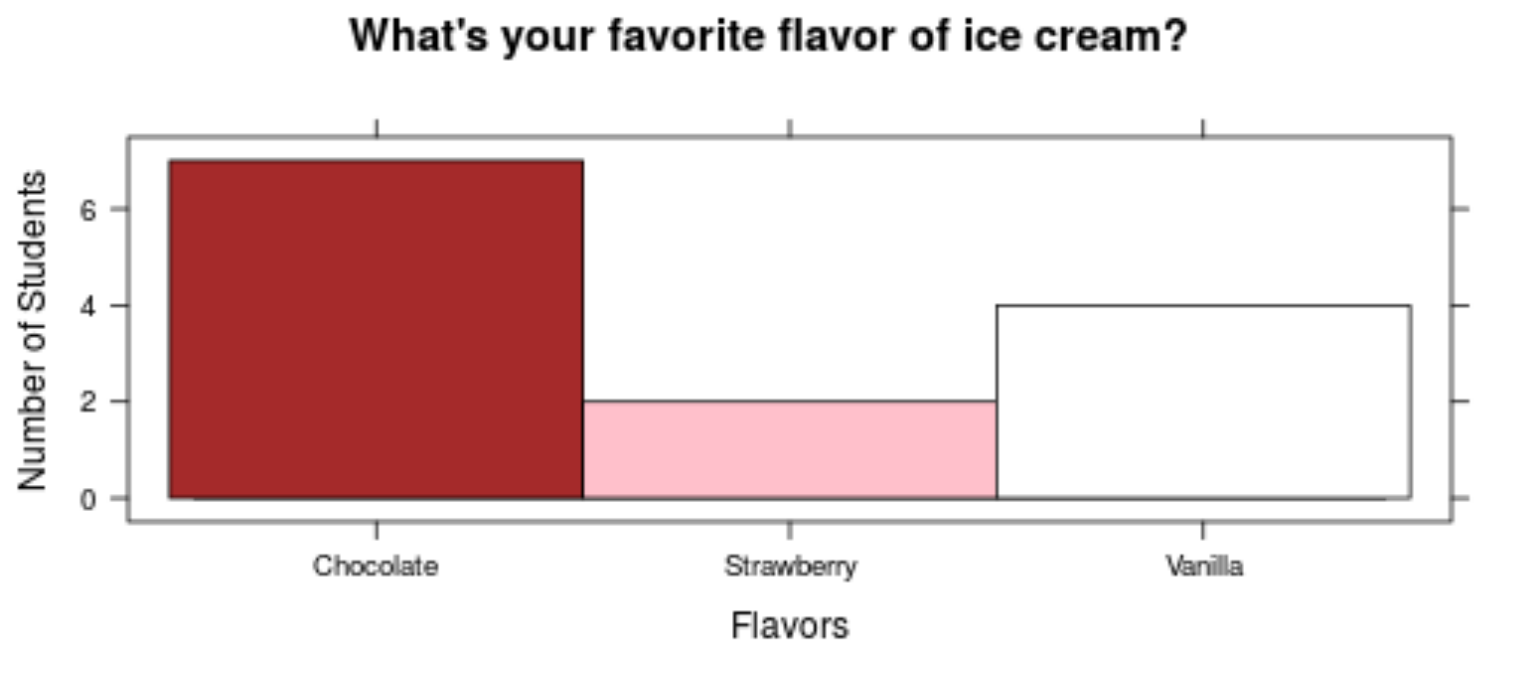
Background & History

- Statistics is “the science of collecting, analyzing, and interpreting numerical data relating to an aggregate of individuals.”
- First used for population and trade purposes
- Computers propelled statistical advancement
- Punch card tabulators were invented in 1890; had widespread use by the 1920s
- In U.S., the first statistical work was done in small labs
  - University of Michigan
  - Iowa State College
    - John Atansoff, inventor of the first electronic computer
- Shift towards personal computing after WWII

The Rise of Statistical Software

- 4 most used statistical programs in scholarly articles: SPSS, SAS, **R** and Stata
- My technical aspect: R
  - Developed by University of Auckland in 1993
    - **R**oss Ihaka & **R**obert Gentleman
  - "Data manipulation, calculation, and graphical display"
  - Open source project & completely free
  - Command-line interface
  - Both a language and an environment
  - Large user community; user-created libraries & packages

I scream, you scream...




```
18 ### Visualization
19
20 ```{r}
21 colors = c("brown", "pink", "white")
22 histogram(datacsv$flavor, main="What's your favorite flavor of ice cream?",
   xlab="Flavors", ylab="Number of Students", col=colors, type="count")
23
```

Significance & Conclusion

- More than just ice cream flavors!
- Statistics affects all fields
- Real-word applications
  - Like psychology!
- Future directions:
  - Create new randomized variables, “color” and “number”
  - See difference between how R reacts to quantitative and categorical variables
  - Conduct analysis with it using techniques from Multiple Regression
  - Experiment with other forms of visualization

References

- 1. “Quotes in Statistics & Science.” *Department of Statistics: University of Wisconsin-Madison*. https://www.stat.wisc.edu/quotes
- 2. “OECD Glossary of Statistical Terms.” *Organisation for Economic Co-operation and Development*. 747. (2007) http://ec.europa.eu/eurostat/ramon/coded_files/OECD_glossary_stat_terms.pdf
- 3. Champkin, Julian. “The timeline of statistics,” *Significance*. Last modified January 24, 2014. https://www.statslife.org.uk/history-of-stats-science/1190-the-timeline-of-statistics
- 4. Grier, David Alan. “The Origins of Statistical Computing.” *Statisticians in History*. http://www.amstat.org/about/statisticiansinhistory/index.cfm?fuseaction=paperinfo&PaperID=4
- 5. Berry, Kenneth J., Johnston, Janis E., & Mielke, Paul W. Jr. *A Chronicle of Permutation Statistical Methods: 1920-2000, and Beyond*. New York: Springer, 2014. http://link.springer.com/book/10.1007/978-3-319-02744-9
- 6. De Leeuw, Jan. “Statistical Software — Overview.” *Department of Statistics, UCLA*. (2009). http://escholarship.org/uc/item/06h5156t
- 7. Muenchen, Robert A. “The Popularity of Data Analysis Sofware.” *rstats.com*. Last modified October 17, 2015. http://r4stats.com/articles/popularity/