

Needles in the Haystack: Google and Other Brokers in the Bits Bazaar

Derived from Blown To Bits Chapter 4
with the same title

CS Concepts

- search engine
- web crawler
- page rank
- primary vs secondary memory access time
- binary search
- who pays for “free” searching

Social Issues

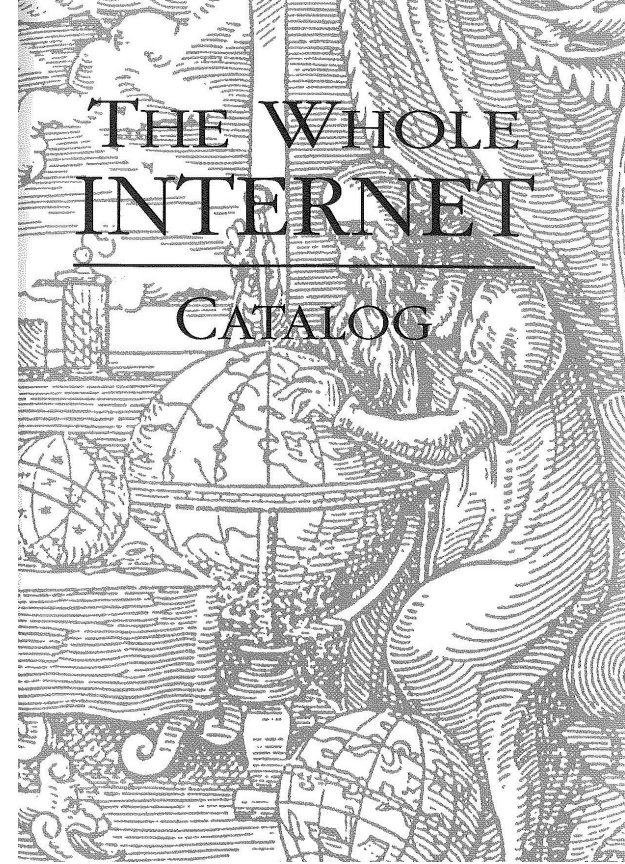
- search as censor
- right to be forgotten
- can't take it back
- free speech
- regulate search engines?

The Library and the Bazaar

- “Yellow pages”, directories, and catalogues
- The “Web” is not hierarchical (no structure like a library)
- Catalogues are out - search engines are in.
- But - search engines control what you see

- How can a search engine respond so fast?
- Does it find every relevant link?
- How does a search engine decide what gets listed first?
- If you try another search engine will you get the same result? If so, which is right? Which is better? Which is more authoritative?
- Are sponsored links better than “organic” links? Is the advertising necessary?
- What is the role of government? What should it be?

The Fall of Hierarchy



- Early catalogues both on the Internet (even before the WWW) and in print.
- “The Whole Internet Catalog” (1994)
- Yahoo was one compiled by human editors.
- Search engines started to be used in early 90s with the growing popularity of the web.

Which of these is an example of information that is NOT hierarchical?

- A. A university course catalogue.
- B. Classification of animals (family, genus, species, etc.)
- C. Telephone numbers.
- D. The location of books in a book store.
- E. Your network of friends.

It Matters How It Works

1. Gather information.
2. Keep copies.
3. Build an index.
4. Understand the query.
5. Determine the relevance of each possible result to the query.
6. Determine the ranking of the relevant results.
7. Present the results.

1. Gather Information

- Spiders or web crawlers wander the web building indices
- estimates range from .02% to 3% of information is indexed
- How often does a page get visited?
 - some frequently (daily), others rarely (determined by the crawler to not be changing)
- How does the crawler find it's way and not go in circles?
- Login's keep bots/crawlers out.

2. Keep Copies

- Spider downloads the page as part of the “visit” in order to create the index.
- Search engine may “cache” the copy.
- Is this legal? What about copyright?
- But wait, browsing requires copying as well.

NYT “By [CLAIRE CAIN MILLER](#)

Published: October 4, 2012

SAN FRANCISCO — After seven years of litigation, Google and book publishers said on Thursday that they had reached a settlement to allow publishers to choose whether Google digitizes their books and journals....”

3. Build an Index

- list of terms and for each term a list of where it appeared
- more than just the terms
 - terms in bigger font might be more important
 - terms in the title might be more important
- must be very fast to lookup
- could be millions of entries (not just words, but names, special numbers, etc.) requiring Gigabytes of memory
- must fit in the computers memory (see next slide)

Binary Search

- Pick a number between 1 and 1000. How many guesses will I need?
- What about that index with 25 million entries?
- The search happens in memory, but the list of URLs associated with the “term” will likely be on disk.

4. Understand the Query

- Steps 1-3 happen in “the background”
- Not much “understanding” in today’s search engine’s but that could change soon.
- Advanced search engine features help

Cardinal’s beat Rangers

vs

Ranger’s beat Cardinal’s

What about a business called “THE”?

5. Determine Relevance

- “Recall” - what percentage of relevant documents are returned by the search?
- Simple relevance calculation -
 - count the number of times each search word appears in the document, add them all up
- Long documents get higher scores.
- Uninteresting words like “the” contribute to the score.
- All word occurrences are not equal (title words should count more).

6. Determine Ranking

- Which of the relevant documents should be displayed first?
- Simple solution - put one with highest relevance score first.
- What if many have the same score?
- Are ones with the highest relevance score really the most important? What about the source of the document (e.g. NY Times vs some random blog post).

Page Rank Algorithms

- Term “Page Rank” comes from Larry Page
- The “crown jewels” of search engines lie in their page rank algorithms.
- Factors include:
 - keywords in heading or titles
 - keyword only in the body text
 - site is “trustworthy”
 - links on this page are to relevant pages
 - links to this page are relevant
 - age of the page
 - quality of the text (e.g. absence of misspellings)

Google' s PageRank Algorithm

- Term PageRank comes from Larry Page (co-founder of Google)
- If lots of pages point TO this page, this must be a “more important” page
- Tweaking the page rank algorithm can make or break a small business.

7. Presenting Results

- Mostly just a list.
- Maybe there are better forms.
- Those sponsored links...

What must fit in computer memory in order to give quick search results?

- A. The index (list) of terms (words etc.) and where on disk to find the pages that match each of those terms.
- B. The list of all URLs known to the search engine and their associated search terms.
- C. Neither A nor B fit entirely in the computers main memory.
- D. Both A and B need to fit to get good performance.

HCI, Robots, Online Education

- Human Computer Interfaces/Interaction
- Robots everywhere?
- Make Sure To Re-watch Videos Shown In Class, As Well As Look At The Questions Provided In The Slides. The Urls Are provided in a file called HCI_Links.pdf on the course webpage, and in ecommons Resources/Week 10 Lecture Slides

BTB Ch 5

CS Concepts

- Caesar cipher
- Vigenere cipher
- One-time pad
- public-key encryption
- one-way computation
- digital signature
- RFID
- WEP, WPA, PGP, RSA
- back door

Social Issues

- legislating back doors
- untappable communications
- internet commerce
- consumer apathy about communication privacy

Simple Encryption Scheme

- Caesar Cipher

GDKKN

- Shift each letter a fixed distance (right or left in the alphabet).

ABCDEFGHIJKLMNOPQRSTUVWXYZ

Simple Encryption Scheme

- Caesar Cipher

WKH DQVZHU LV G

- Decode this encrypted message and you will know which answer to pick.

ABCDEFGHIJKLMNOPQRSTUVWXYZ

Better Encryption Scheme

- Substitution Cipher

ABCDEFGHIJKLMNOPQRSTUVWXYZ

ZWKBJLSIHCFGRA XEMNTUQPDOVY

- $26!$ or about 4×10^{26} different encoding/keys
- Caesar had just 26 – could just try them all

Decode This

- Key

ABCDEFGHIJKLMNOPQRSTUVWXYZ

ZWKBJLSIHCFGRAXEMNTUQPDOVY

JXQVTSPM

- A. Synonym for laptop
- B. Synonym for school
- C. Synonym for house
- D. Synonym for pencil
- E. Synonym for table

Vigenère Cipher

- Cycle through a set of Caesar ciphers to combat frequency analysis.

ABCDEFGHIJKLMNOPQRSTUVWXYZ

NOPQRSTUVWXYZABCDEFGHIJKLM

DEFGHIJKLMNOPQRSTUVWXYZABC

. . .

- Key is the left column, which can be used to reconstruct the table.

Vigenère Cipher

ABCDEFGHIJKLMNOPQRSTUVWXYZ

NOPQRSTUVWXYZABCDEFGHIJKLM

DEFGHIJKLMNOPQRSTUVWXYZABC

HJKLMNOPQRSTUVWXYZABCDEFGHI

- How many letters must be transmitted to transmit the “key” to this Vigenère cipher?

A. 3

B. 4

C. $26 \times 3 = 78$

D. $26 \times 4 = 104$

Vigenère Cipher

ABCDEFGHIJKLMNOPQRSTUVWXYZ

NOPQRSTUVWXYZABCDEFGHIJKLM

DEFGHIJKLMNOPQRSTUVWXYZABC

HJKLMNOPQRSTUVWXYZABCDEFGHI

- Decode this: “ODUNQH”

A. BANDED

B. BANANA

C. BANGLE

D. BANTER

One-time Pad

- If the Vigenère key is as long as the text then it is called a one-time pad.
- Unbreakable in theory.

Vigenère Cipher

- “The amount of computation required to break a cipher by exhaustive search grows exponentially in the size of the key. Increasing the key length by one bit doubles the amount of work required to break the cipher, but only slightly increases the work required to encrypt and decrypt.”
 - BTB pg. 173

Public Key Cryptography

- Pre-public key, assume the message will be intercepted so make it unreadable.
- Required the two parties to have a shared key.
- Breakthrough allows parties to create a key using a non-private communication, without ever meeting.

Diffie and Hellman

- didn't need to have a shared secret (the key) in advance to communicate securely
- uses one-way computation and
- a key agreement protocol

Key Agreement Protocol

- $X * Y$ is easy but finding X given Y is HARD (not $*$ is NOT multiplication)
- $X * Y * Z = X * Z * Y$
- everybody knows how to do $X * Y$ and everybody knows some number g

Key Agreement Protocol

$$K = a * B = a * g * b$$

$$K = b * A = b * g * a$$

A

B

a

$$A = g * a$$

b

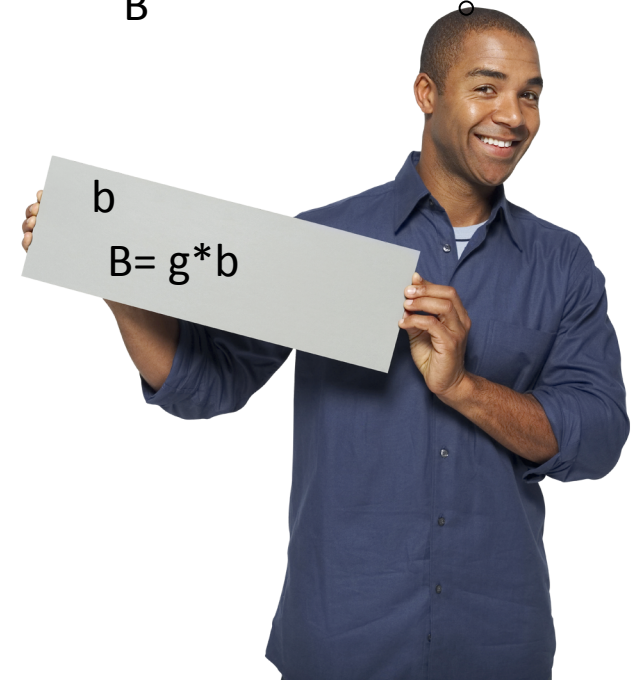
$$B = g * b$$

Eaves Dropper Knows:

A

B

g



Public Key Messaging

$$K = a * B = a * g * b$$

$$K = b * A = b * g * a$$

A

B + msg:K

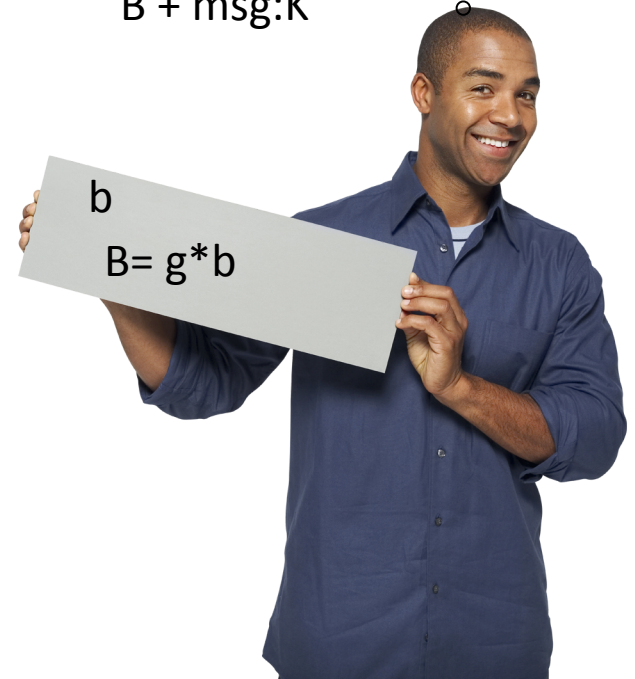
Public Directory

a

$$A = g * a$$

b

$$B = g * b$$



A Shared Secret?

- A. Using Public Key cryptography, one of the two communicating parties must first **SECRETLY** communicate a special “key” to the other that they then use to carry on the rest of their secret communication.
- B. Using Public Key cryptography, **ALL** messages passed between the two parties can be read by a third party without risk of the secret communication being compromised.

Digital Signatures

- Send plain text message along with essentially an encrypted copy of the message as the “signature”.
- System relies on the fact that it is easy to encrypt a message with a private key (e.g. Alice’s a) and easy to decrypt a message with a public key (e.g. Alice’s A).
- Bob decrypts the “signature” with A and if he doesn’t get the original message back, then it was tampered with in some way.

RSA – Rivest, Shamir, and Adleman

- Found a way to compute private-public key pairs (e.g. a and A) such that they perform inverse operations.
- $\text{RSA}(\text{msg}, a) = \text{secret}$
- $\text{RSA}(\text{secret}, A) = \text{msg}$

RSA Encryption

- https://www.youtube.com/watch?v=wXB-V_Keiu8

Is that REALLY Alice's public key?

- How do you know you are really talking to Alice, or that was her key you found in that public directory?
- Enter certificates and certificate authorities.

Snap Practice

Assume nouns initially ends with salami what does fiddle end with after executing these blocks?

set fiddle to all but first of nouns

set dee to fiddle

add thing to nouns

add other to fiddle

- A. salami
- B. salami, thing
- C. salami, thing, other
- D. salami, other

Assume nouns initially ends with salami what does dee end with after executing these blocks?

A stack of four orange Scratch code blocks. The first block is a 'set' block with 'fiddle' in the variable dropdown, 'to', 'all but first of', and 'nouns' in the value dropdown. The second block is a 'set' block with 'dee' in the variable dropdown, 'to', and 'fiddle' in the value dropdown. The third block is an 'add' block with 'thing' in the text input, 'to', and 'nouns' in the variable dropdown. The fourth block is an 'add' block with 'other' in the text input, 'to', and 'fiddle' in the variable dropdown.

set fiddle ▾ to all but first of nouns

set dee ▾ to fiddle

add thing to nouns

add other to fiddle

- A. salami
- B. salami, thing
- C. salami, thing, other
- D. salami, other

Recursion

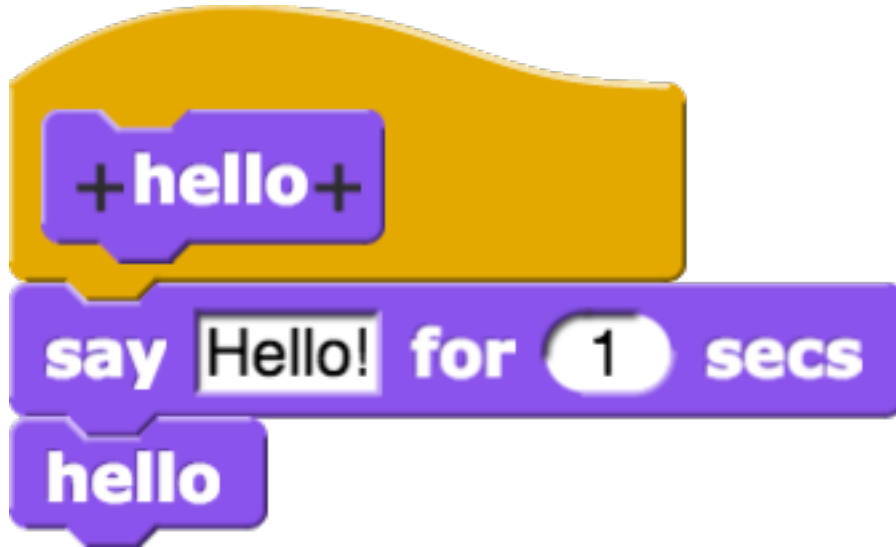
- A block that calls itself
- recursion
 - recursion
 - recursion
 - recursion
 - » recursion
 - recursion
 - recursion
 - recursion

UPPER CRUST WELCOMES YOU TO SANTA CRUZ



PIZZA
of the
SUCE

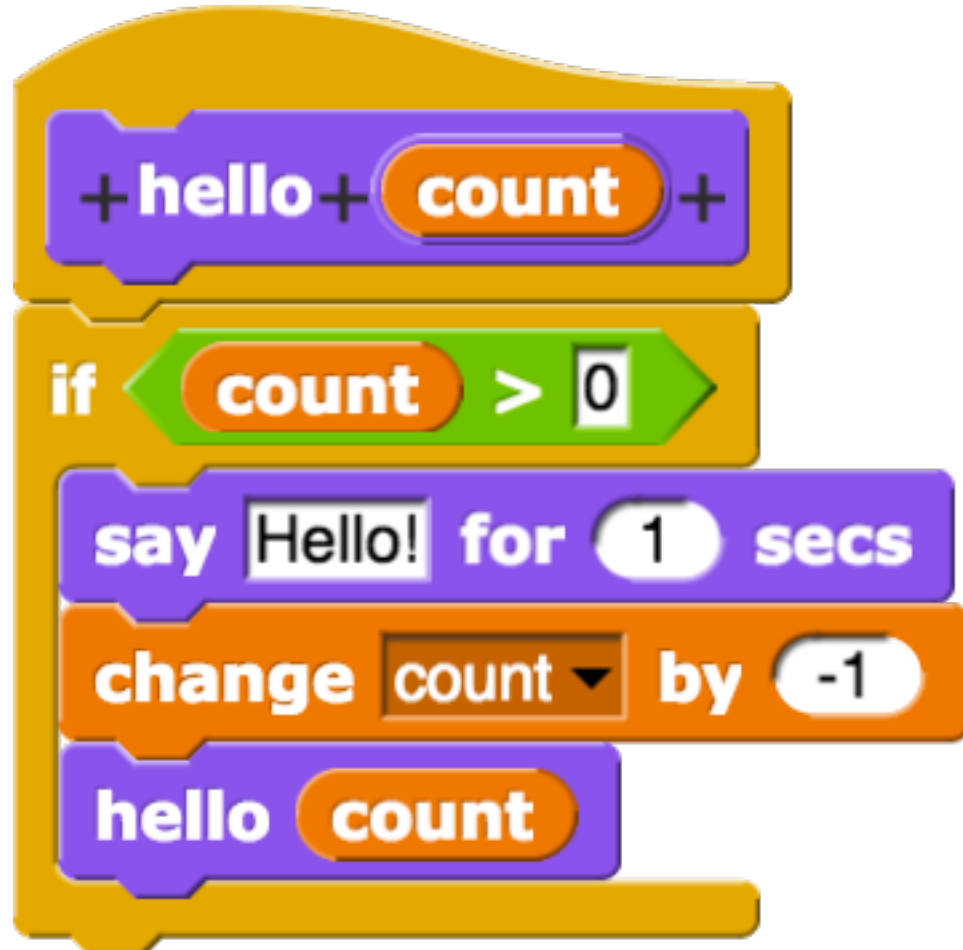
For how many seconds does this say “Hello!”?



- A. 0
- B. 1
- C. forever

For how many seconds does this say “Hello!”?

hello 3



- A. 1
- B. 2
- C. 3
- D. 4
- E. forever

What is printed by this block?

mystery 4



```
+mystery + x +  
If x > 1  
  report x * mystery * x - 1  
else  
  report 1
```

- A. 12
- B. 16
- C. 20
- D. 24

Progress has been tremendous

Artificial Intelligence (is no match
for natural stupidity :)

*Lawrence Snyder
University of Washington, Seattle*

Thinking with Electricity

- The inventors of ENIAC, 1st computer, said it “thinks with electricity”
 - Do calculators “think”?
 - Does performing arithmetic, which is entirely algorithmic, require thinking?
 - Once, performing arithmetic, was thought to be a divinely or magically conferred ability

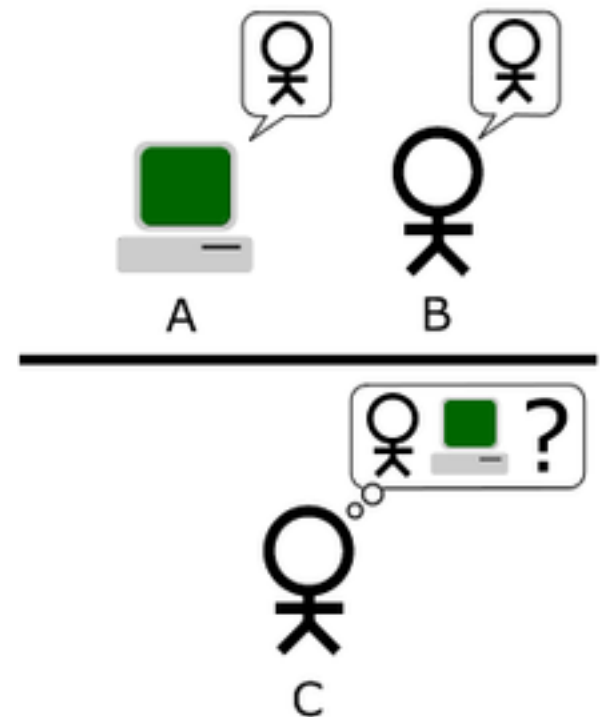
The Problem: Many human activities look like thinking until they are understood (to be algorithmic)

Turing's Test

- A.M. Turing, computer pioneer, worried about intelligence in humans & machines; proposed a test (1950)

- Aware that it is intelligence until it's understood, Turing devised a test with this experimental setup:

A computer (A) and a person (B) are connected to an examiner, who can ask questions using text to determine which is which

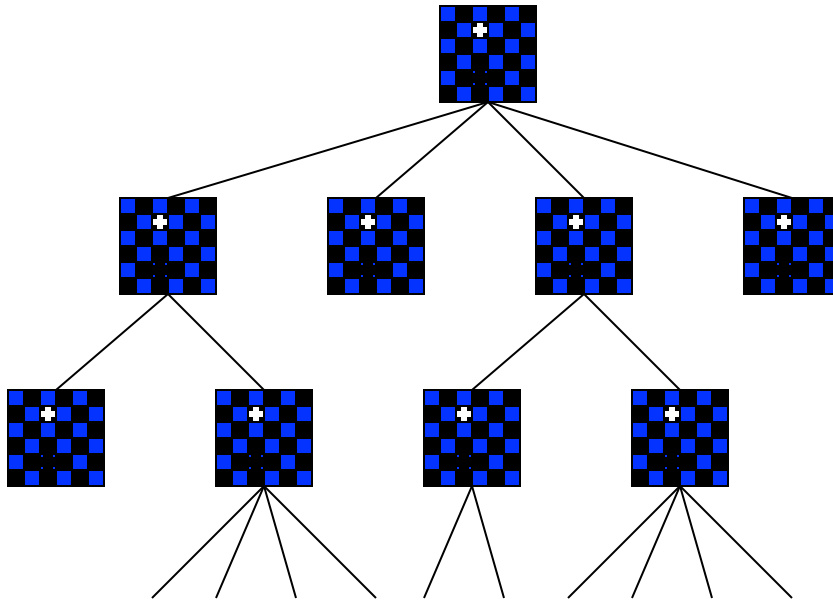


Artificial Intelligence

- The study of making computers act intelligently
 - They already act intelligent ... e.g. they can correct your spelling mistakes
 - Is this intelligent behavior? Most AI researchers would say “no” ... algorithmic
 - Playing grandmaster level chess in a tournament became an AI goal (1952)
 - Minimizes real world knowledge
 - Clear goal, formal system

Playing Chess

- Chess is a game, so it uses a game tree
 - At each node is a ‘board’ -- easily digitized
 - Below it are all boards created in 1 move; below those, all boards needing another move



An objective function evaluates “goodness” of the position: go for highest ... opponent goes for lowest

Deep Blue vs Kasparov

- An IBM system, Deep Blue, played world champion Gary Kasparov
 - In 1996 Kasparov won, but Deep Blue played 1 game well!!! This was a first.
 - In May 11, 1997 Deep Blue won 3.5-2.5

Deep Blue is a 32 processor parallel computer with 256 “chess processors” that can consider 200,000,000 chess positions per second + opens + ends

Intelligent?

- Does Deep Blue's performance show that a computer can be intelligent?
 - No -- it repeats its designers intelligence (weak rebuttal)
 - Yes -- it's better than anyone in the world at something people find interesting and fun
 - Maybe -- it shows intelligence in chess, but can it apply its intelligence elsewhere?

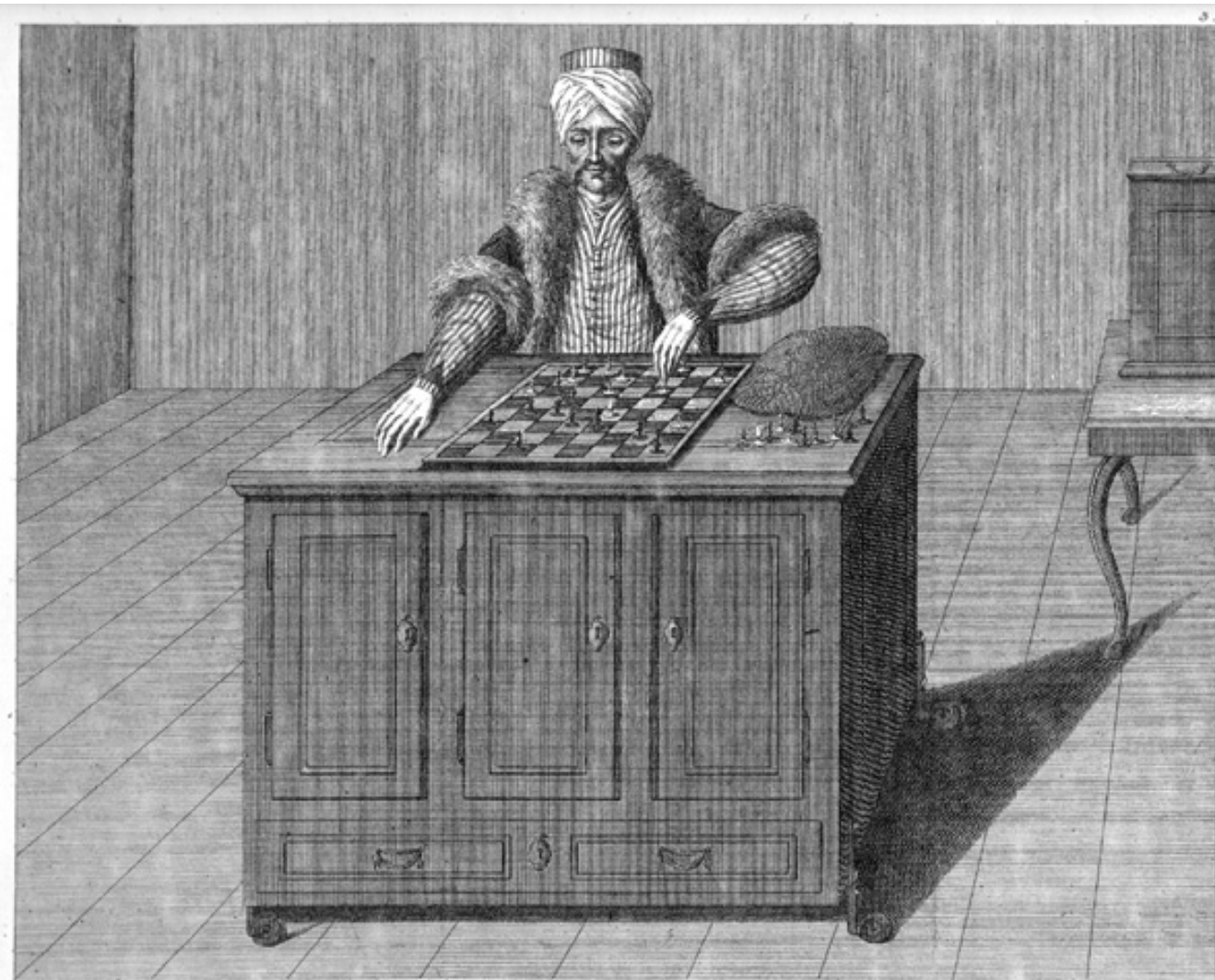
What do you think?

Chinese Room – John Searle



Photo "ChineseRoom2009 CRset" by TheChineseRoom

The Turk



W. de Kempelen del.

Chr. à Meckel excud. Basilea.

P. G. Pintz. sc.

Der Schachspieler im Spiele begriffen. Le Joueur d'Échecs tel qu'on le voit pendant le jeu.

And Now Watson plays Jeopardy



Watson: More Sophisticated

- Compared to Deep Blue, Watson is much more sophisticated in design, organization
 - runs on ~2,500 parallel CPUs, each capable of up to 33 billion operations a second; size of small RV
 - crawled and organized 200 million pages of data
 - “expert” analyzers — more than 100 different techniques running concurrently to analyze natural language, appraise sources, propose hypotheses, merge results and rank top guesses.

Compare Watson, Deep Blue

- Chess seems harder, but it's not
 - Chess has fixed rules, little real world data needed
 - Jeopardy, more free form using only real data
- Other differences
 - In chess the “problem” is known beforehand, but in Jeopardy, someone else sets up the problem
 - In chess, decisions are based on a formula, but in Jeopardy many forms of evaluation are needed (a problem solved by probabilities)
 - In chess there is very little pre-planning, but in Jeopardy, organizing the data is the key

Composing Music

- An experiment at the U. of Oregon ... compose music in the style of Bach
- Three participants: Bach, U of O Professor, EMI program
- And the winner is ...

Audience Thought:	Bach's	Professor's	EMI's
	work was	work was	work was
	Professor	EMI program	Bach

Other AI/Machine Learning Questions

- Robotics
- Facial recognition
- Route planning
- Displaying emotions
- Natural Language Processing
- Pattern recognition/prediction – medicine, weather, economics, spam
- Recommender systems
- Image browsers

Connected computers are better! How's it done?

Networking ...

*Based on slides by Lawrence Snyder
University of Washington, Seattle*

Networks...

Computers are useful alone, but are better when connected (networked)

- Access more information and software than is stored locally
- Help users to communicate, exchange information...changing ideas about social interaction
- Perform other services—printing, Web, email, texting, mobile, etc.

Network Structure

Networks are structured differently based (mostly) on distance between computers:

- Local area network (LAN)
 - Small area: room or building
 - Either wired (Cu or fiber) or wireless
- Wide area networks (WAN)
 - Large area: more than 1 km
 - Fiber-optic, copper transmission lines, μ -wave, satellite
- Metropolitan area networks (MAN)
 - Neighborhood or several blocks of business district
 - Private service provider owns network

Protocol Rules!

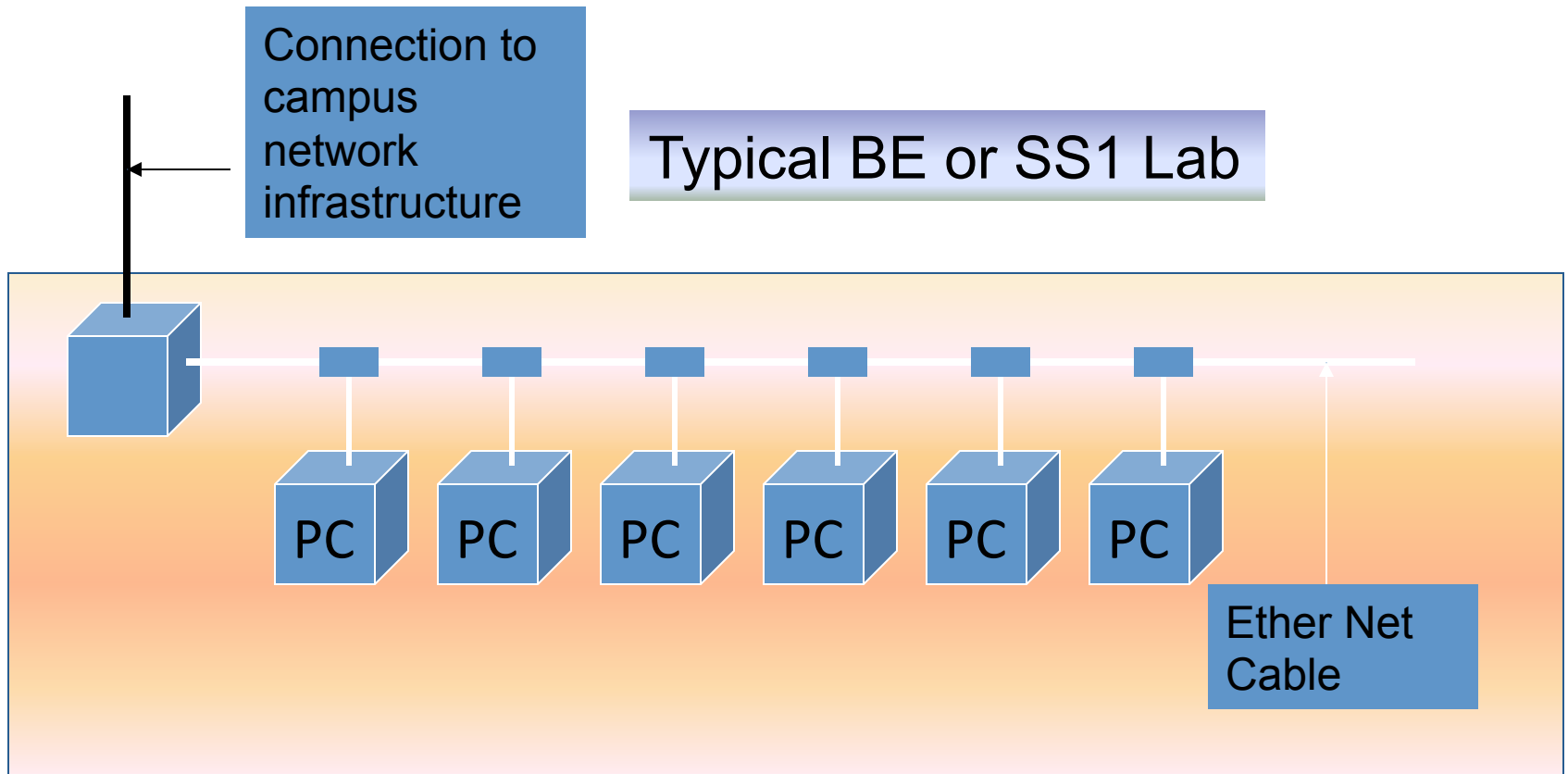
To communicate computers need to know how to set up the info to be sent and interpret the info received

- Communication rules are a *protocol*
- Example protocols
 - EtherNet—for physical connection in a LAN
 - TCP/IP—for Internet—transmission control protocol / internet protocol
 - HTTP—for Web—hypertext transfer protocol

LAN in the Lab

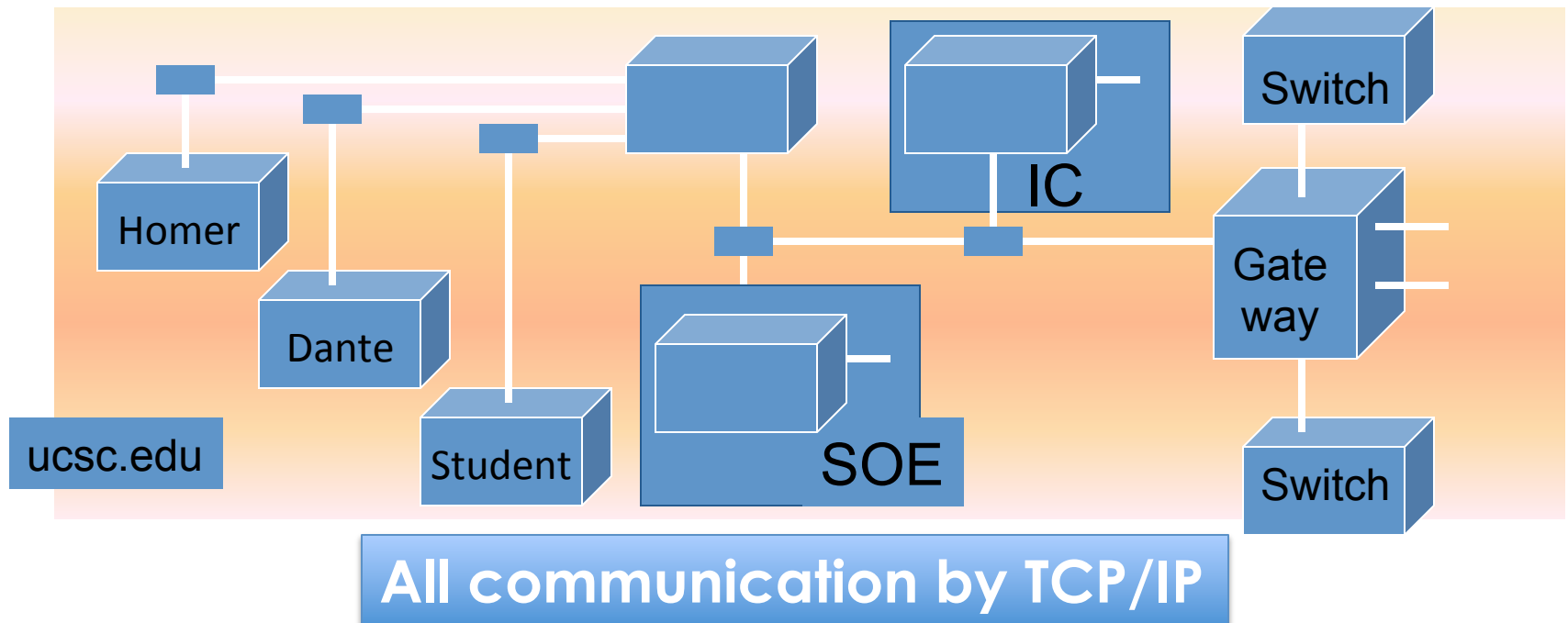
EtherNet is a popular LAN protocol

- It uses a “party” protocol



Campus & The World

The campus subnetworks interconnect computers of the UCSC domain which connects to Internet via a gateway



IP—Like Using Postcards

Information is sent across the Internet using IP—Vint Cerf uses postcard analogy

- Break message into fixed size units
- Form IP packets with destination address, sequence number and content
- Each makes its way separately to destination, possibly taking different routes
- Reassembled at destination forming msg

addr # data

Key Point: Taking separate routes lets packets bypass congestion and out-of-service switches; packet reassembly discovers lost packets; ask for resend

Routes in the Internet

- A. Sending a big message across the internet is like a train with many cars (smaller packets) that are linked together to form a long train (the entire message).
- B. Sending a big message across the internet is like sending a bunch of cars (smaller packets) that each take their own route to the destination and get reassembled after they all get there. Some might not even make it the first time.

Route Across the US

- The route
 - Starts with my ISP moving packet through MAN
 - Next, the packet enters a regional WAN
 - Next, the packet crosses the backbone
 - Arriving at another regional WAN
 - Next arriving on campus in a LAN
 - Delivered to the destination computer

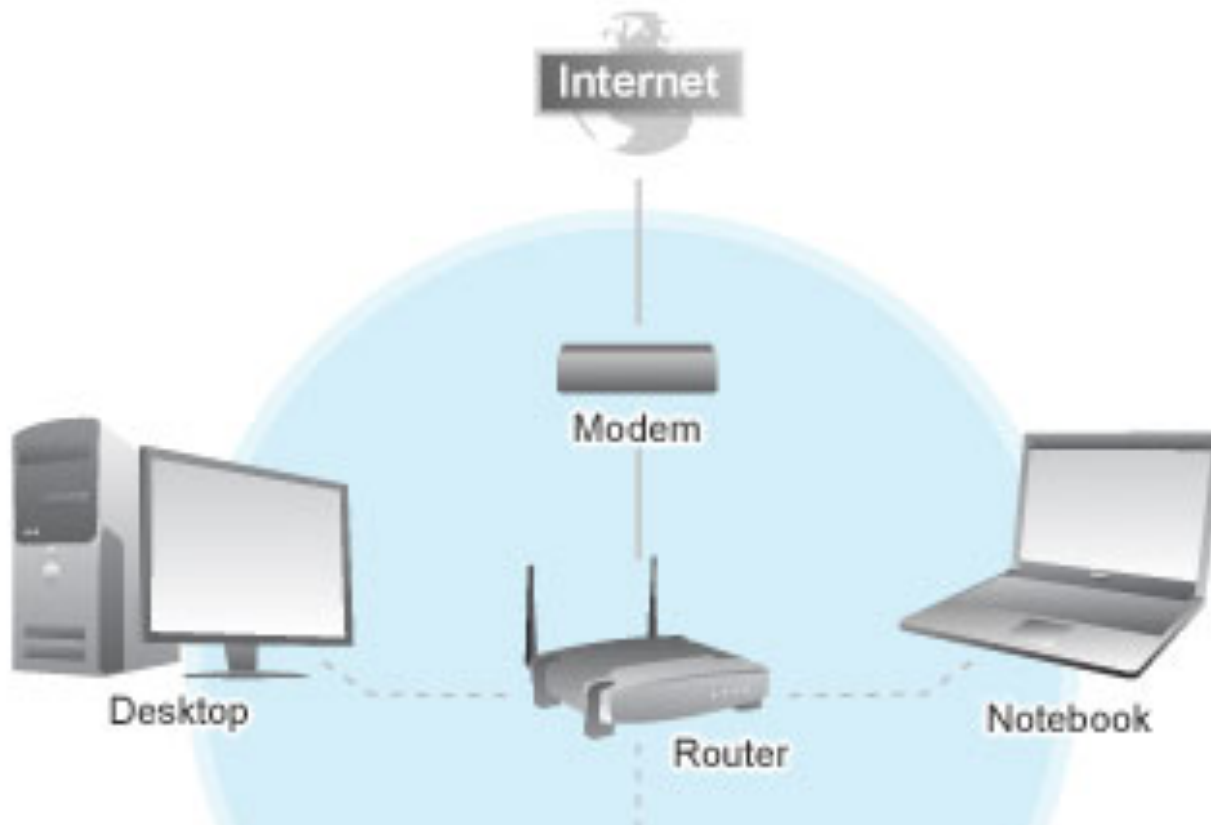
Regional Network

- The Watchtower regional network of Eastern MA
- Every state/region has one or a few



Wireless is a LAN technology

- As with “wired Ethernet,” all computers in range can hear the radio signals of the others



Naming Computers—Take 1

People name computers by a domain name

– a hierarchical scheme that groups like computers

- **.edu** All educational computers, a TLD
- **.ucsc.edu** All computers at UCSC
- **.ic.ucsc.edu** All Instructional Computing computers
- **.soe.ucsc.edu** School of Engr. Computers
- **riverdance.soe.ucsc.edu** An SOE computer
- **unix.ic.ucsc.edu** An IC computer.

Domains begin with a “dot” and get “larger” going right

Naming Computers—Take 2

Computers are named by IP address, four numbers in the range 0-255

unix.ic.ucsc.edu: **128.114.104.230**

riverdance.soe.ucsc.edu: **128.114.48.104**

- Remembering IP addresses would be brutal for humans, so we use domains
- Computers find the IP address for a domain name from the *Domain Name System*—an IP address-book computer

A computer needs to know IP address of DNS server!

Domains

.edu .com .mil .gov .org .net domains are “top level domains” for the US

– Recently, new TLD names added

– Each country has a top level domain name:

- .ca (Canada)
- .es (Spain)
- .de (Germany)
- .au (Australia)
- .at (Austria)
- .us (US)

Do you know sites like:

bit.ly

www.nba.tv

del.icio.us

... they exploit TLDs

Logical vs Physical

View the Internet in two ways:

1. Humans see a hierarchy of domains relating computers—**logical network**
2. Computers see groups of four number IP addresses—**physical network**

Both are ideal for the “user's” needs

- The Domain Name System (DNS) relates the logical network to the physical network by translating domains to IP addresses

True or false, Internet addresses and names are hierarchical?

- A. Both are hierarchical
- B. Neither are hierarchical
- C. Just the names
- D. Just the addresses

Internet vs. World Wide Web

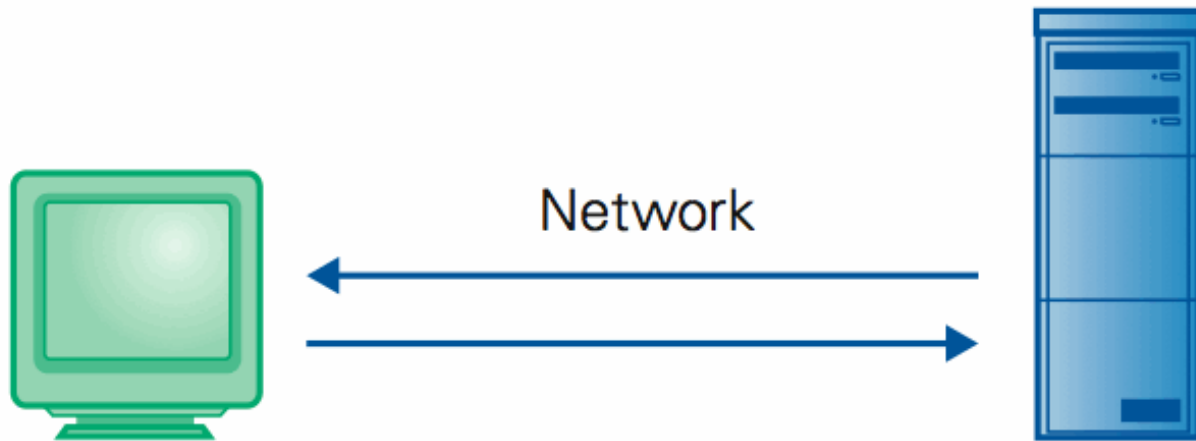
- Many people mis-use the terms “Internet” and “World Wide Web”
- Let’s get them right

Internet: all of the wires, fibers, switches, routers etc. connecting named computers

Web: That part of the Internet —web servers—that store info and serve Web pages and provide other services to client computers

One More Protocol: Client/Server

- The Web and much of the Internet services use the client server form of interaction



Client Computer

Requests services

(Sends URL for a Web page)

Server Computer

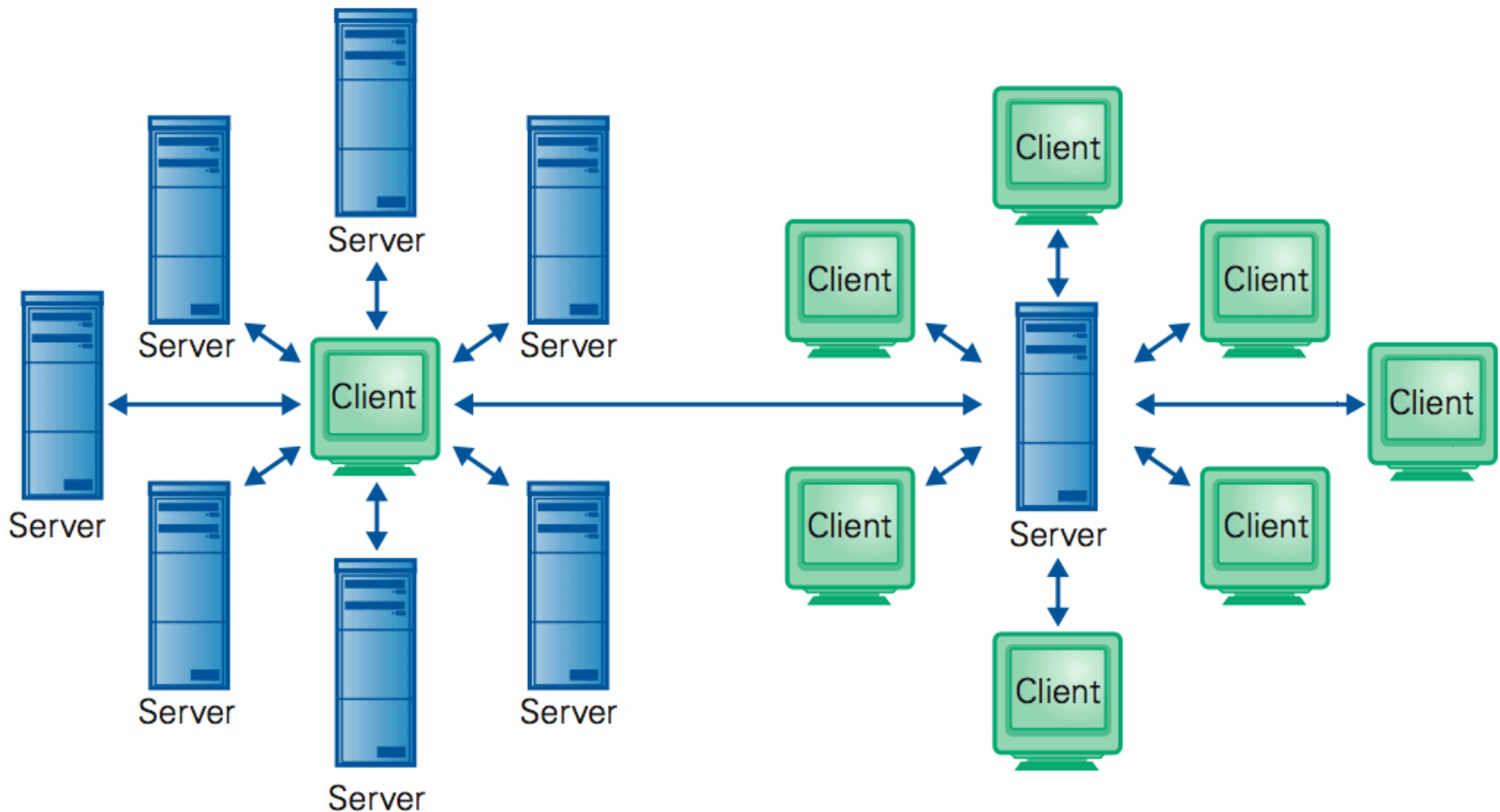
Provides services

(Returns the Web page file)

- It's a VERY BRIEF relationship

Client/Server Is Also Smart

- Clients and servers are not connected – they only exchange info ... “no commitment issues”



Internet ~~is Running~~ out of Addresses

HAS RUN

- The last top level IPv4 addresses were assigned in February 2011.
- IPv4 uses 32 bits and thus can specify 4.3 billion IP addresses.

An IPv4 address (dotted-decimal notation)

172 . 16 . 254 . 1



10101100 ,00010000 ,11111110 ,00000001



One byte =Eight bits

Thirty-two bits (4 x 8), or 4 bytes

IPv6

The decision to put a 32-bit address space on there was the result of a year's battle among a bunch of engineers who couldn't make up their minds about **32, 128, or variable-length**. And after a year of fighting, I said—I'm now at ARPA, I'm running the program, I'm paying for this stuff, I'm using American tax dollars, and I wanted some progress because we didn't know if this was going to work. So I said: OK, it's 32-bits. That's enough for an experiment; **it's 4.3 billion terminations**. Even the Defense Department doesn't need 4.3 billion of everything and couldn't afford to buy 4.3 billion edge devices to do a test anyway. **So at the time I thought we were doing an experiment to prove the technology and that if it worked we'd have opportunity to do a production version of it. Well, it just escaped!** It got out and people started to use it, and then it became a commercial thing. So this [IPv6] is the production attempt at making the network scalable.

—[Vint Cerf](#), *Google IPv6 Conference 2008*

IPv6

- 128 bit addresses
- Approximately 3.4×10^{38} addresses.
- That's 4.8×10^{28} addresses for each of the 7 billion people alive in 2011.

An IPv6 address (in hexadecimal)

2001:0DB8:AC10:FE01:0000:0000:0000:0000



2001:0DB8:AC10:FE01::

Zeroes can be omitted



0010000000000001:0000110110111000:1010110000010000:1111111000000001:

0000000000000000:0000000000000000:0000000000000000:0000000000000000

Which item does not belong in this list?

- A. Ethernet
- B. ftp
- C. http
- D. html
- E. X.25

Relating the “logical” with the “physical”

Domain Name System

Lawrence Snyder
University of Washington, Seattle

Recall 2 Ways To Name Computers

- **Logical:** Humans use domain names
 - riverdance.soe.ucsc.edu
- **Physical:** Computers use number-quads
 - 128.114.48.104
- This is different than the phone system:
 - The people use numbers: 1 800 555 1212
 - The equipment uses the same numbers
- A key property of computers: they can separate the logical form (preferred by people) from the physical form they must use

Today ...

- Today, we explain how the logical/physical separation is implemented for domain names
- But, this is also a chance to illustrate the structure of LARGE systems
 - Study the basic components
 - Study design ideas that make the system work well
 - This matters to you because you’ ll probably have “big ideas” about using computers

What's the Problem?

- The Internet is completely decentralized
 - No one is in charge – ICANN
 - A few companies get permission to give users or organizations IP-addresses – not much logic to it
 - When a person or organization gets an IP-address, it picks a domain name – few rules
- Once connected to I' net, users start using domain name ... but when someone refers to it, how does their computer get its number?

Internet Corporation for
Assigned Names and Numbers

Recall mail to “friend@cise.ufl.edu”

68.87.205.1	-	Mt Laurel, usa
68.85.240.101	be-70-ar01.burien.wa.seattle.comcast.net	Mt Laurel, usa
68.85.240.69	be-30-ar01.seattle.wa.seattle.comcast.net	Seattle, WA, USA
68.86.90.213	pos-0-5-0-0-cr01.seattle.wa.ibone.comcast.net	Seattle, WA, USA
68.86.85.206	pos-0-8-0-0-cr01.portland.or.ibone.comcast.net	Portland, OR, USA
68.86.85.197	pos-1-15-0-0-cr01.sacramento.ca.ibone.comcast.net	Sacramento, CA, USA
68.86.85.181	pos-0-9-0-0-cr01.sanjose.ca.ibone.comcast.net	San Jose, CA, USA
154.54.11.105	te3-3.mpd01.sjc04.atlas.cogentco.com	San Jose, CA, USA
154.54.0.177	te9-1.ccr02.sfo01.atlas.cogentco.com	San Francisco, CA, USA
154.54.3.137	te3-8.ccr01.lax01.atlas.cogentco.com	Los Angeles, CA, USA
154.54.0.226	te3-8.ccr01.iah01.atlas.cogentco.com	Houston, TX, USA
154.54.24.194	te3-2.ccr01.mia01.atlas.cogentco.com	Miami, FL, USA
154.54.1.186	te3-3.ccr01.mia03.atlas.cogentco.com	Miami, FL, USA
38.112.31.66	florida_lambda_rail_llc.demarc.cogentco.com	Washington, DC, USA
198.32.155.10	tpa-flrcore-7609-1-te21-1.net.flrnet.org	Marina del Rey, usa
198.32.173.161	tlh-flrcore-7609-1-te41-1907.net.flrnet.org	Marina del Rey, usa
198.32.173.162	ctx36-ewan-msfc-1-v1907-1.ns.ufl.edu	Marina del Rey, usa
128.227.236.85	ctx36-nexus-msfc-1-v50-1.ns.ufl.edu	Gainesville, FL, USA
128.227.236.14	csev1-core-msfc-1-v41-1.ns.ufl.edu	Gainesville, FL, USA
128.227.254.74	-	Gainesville, FL, USA
128.227.205.2	cise.ufl.edu	Gainesville, FL, USA

But, how do we get 128.227.205.2?

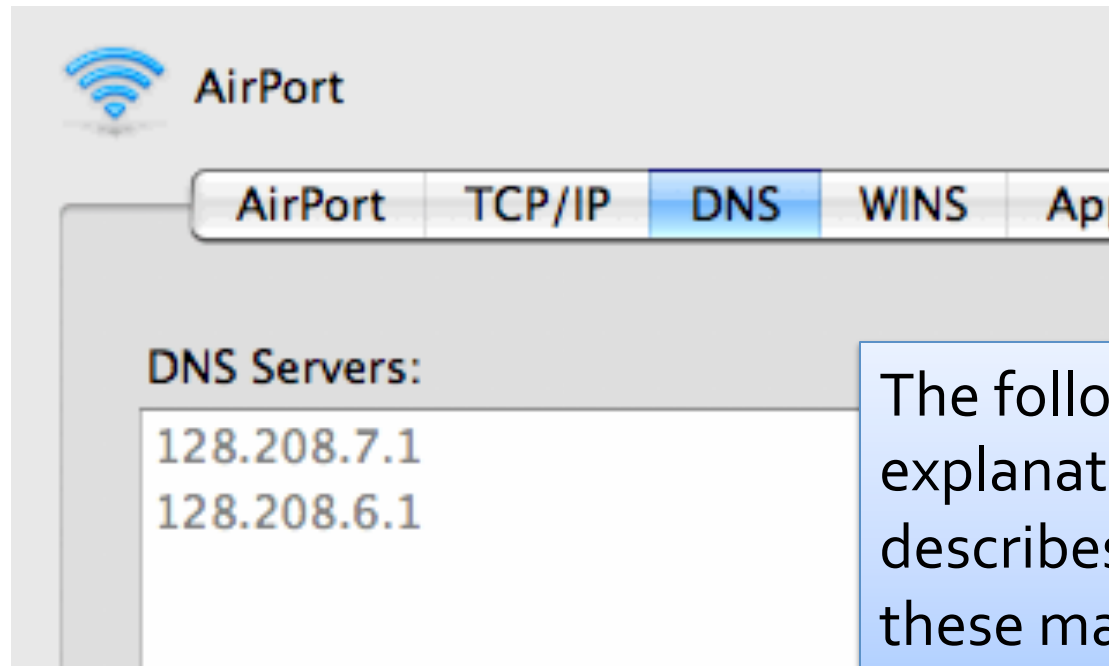
- When we send mail to a friend at the U of FL, we type friend@cise.ufl.edu and the computer that sends mail for us on campus needs to find out this fact:

cise.ufl.edu == 128.227.205.2

- We said it asks the Domain Name System, or DNS ... so what happens

But Wait!

- How does it know the address of the DNS?
- You (or someone or something who set up your computer) told it when connecting it to the network ... look in net control panel



The following explanation describes what these machines do

First Step

- The DNS server answers the question “what number is **cise.ufl.edu**?” by this method
- First Step: Look it up in the “address book”
 - The DNS server does that
 - It keeps its own address book, a list of all of the domain names like **cise.ufl.edu** that it has been asked about and found
 - We say it *caches* the addresses it’ s found
 - *caching – keeping a copy around in case need it again*
 - It checks the cache first

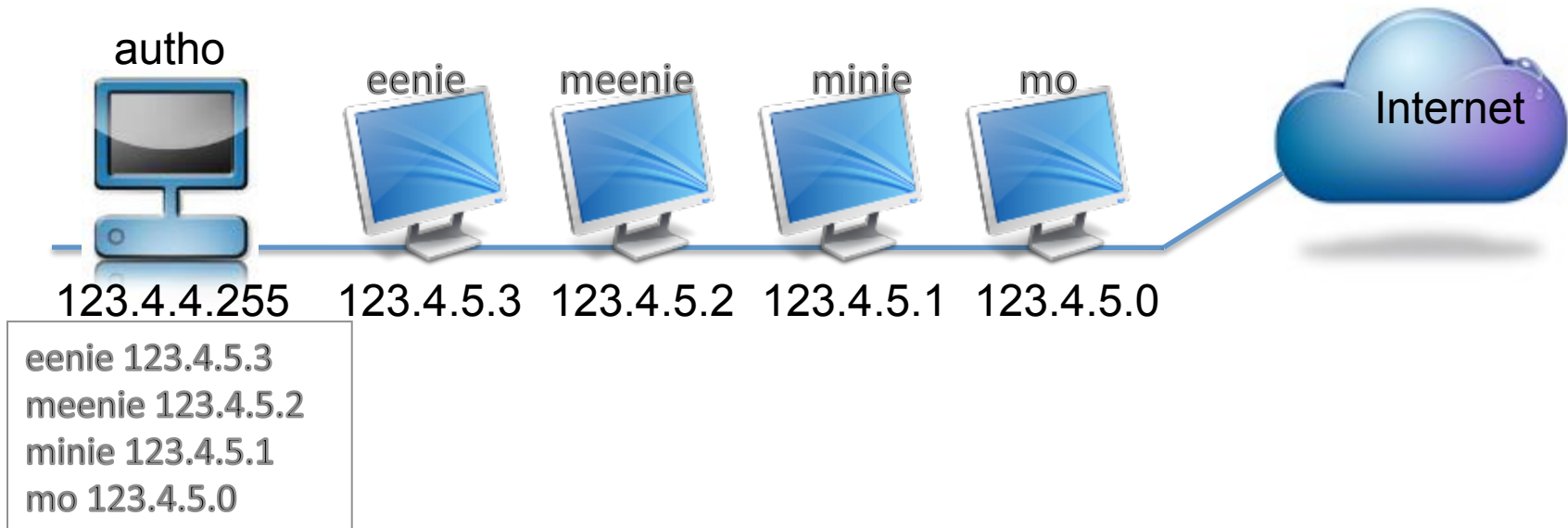
If It Has Never Been Asked ...

- The address will not be in the cache if this is the first request
- Second Step: The DNS server begins a process of finding the address on behalf of your computer ...

That process uses 2 Facts of I' net

The DNS Design: Fact 1

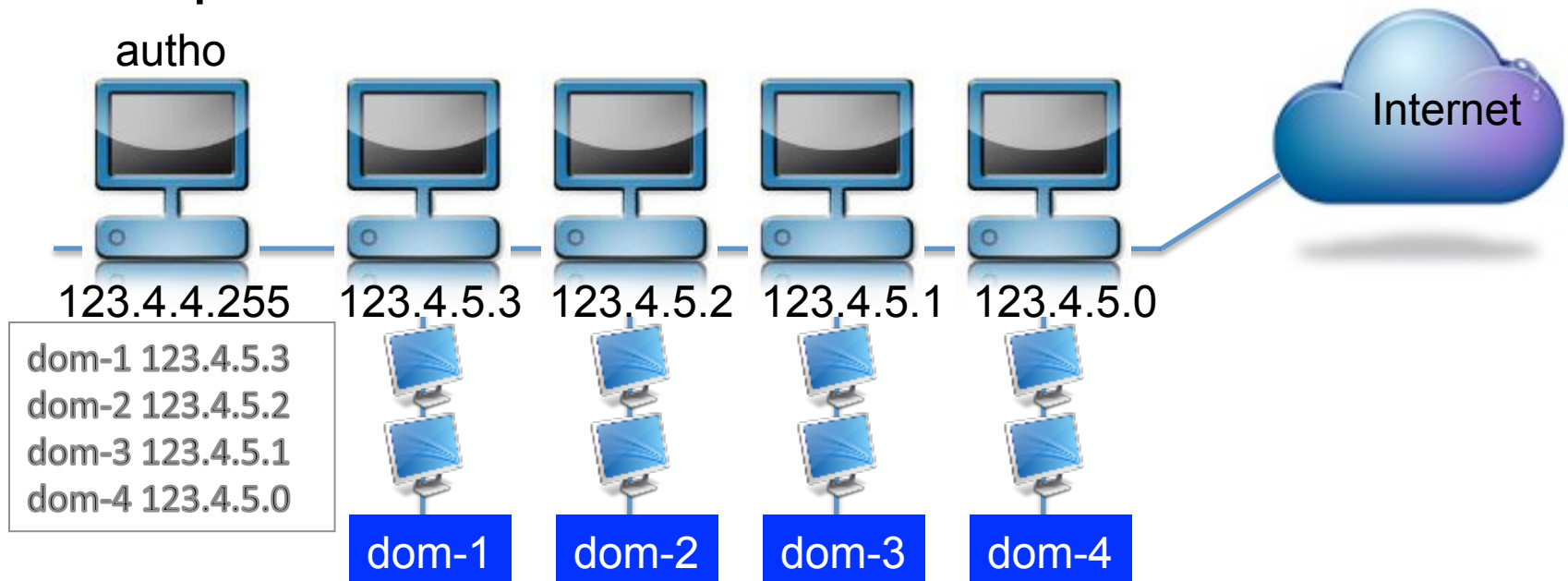
- Every domain has an authoritative name server, which I'll call autho



- Two Cases: Autho knows the number of every computer in its domain

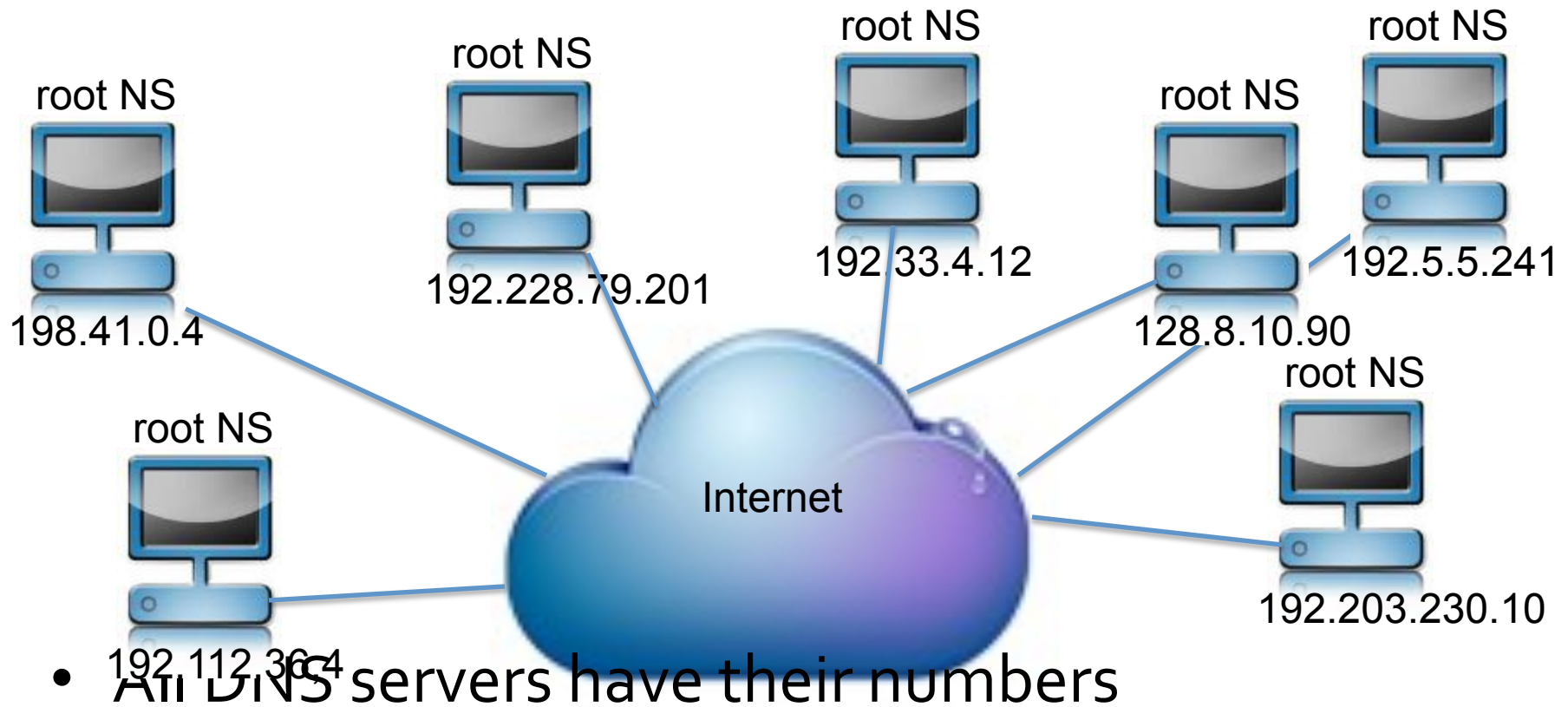
The DNS Design: Fact 1 (Continued)

- OR Autho knows the number of every autho computer in its domain



The DNS Design: Fact 2

- There are 13 Internet “root name servers” scattered around the world ... all the same

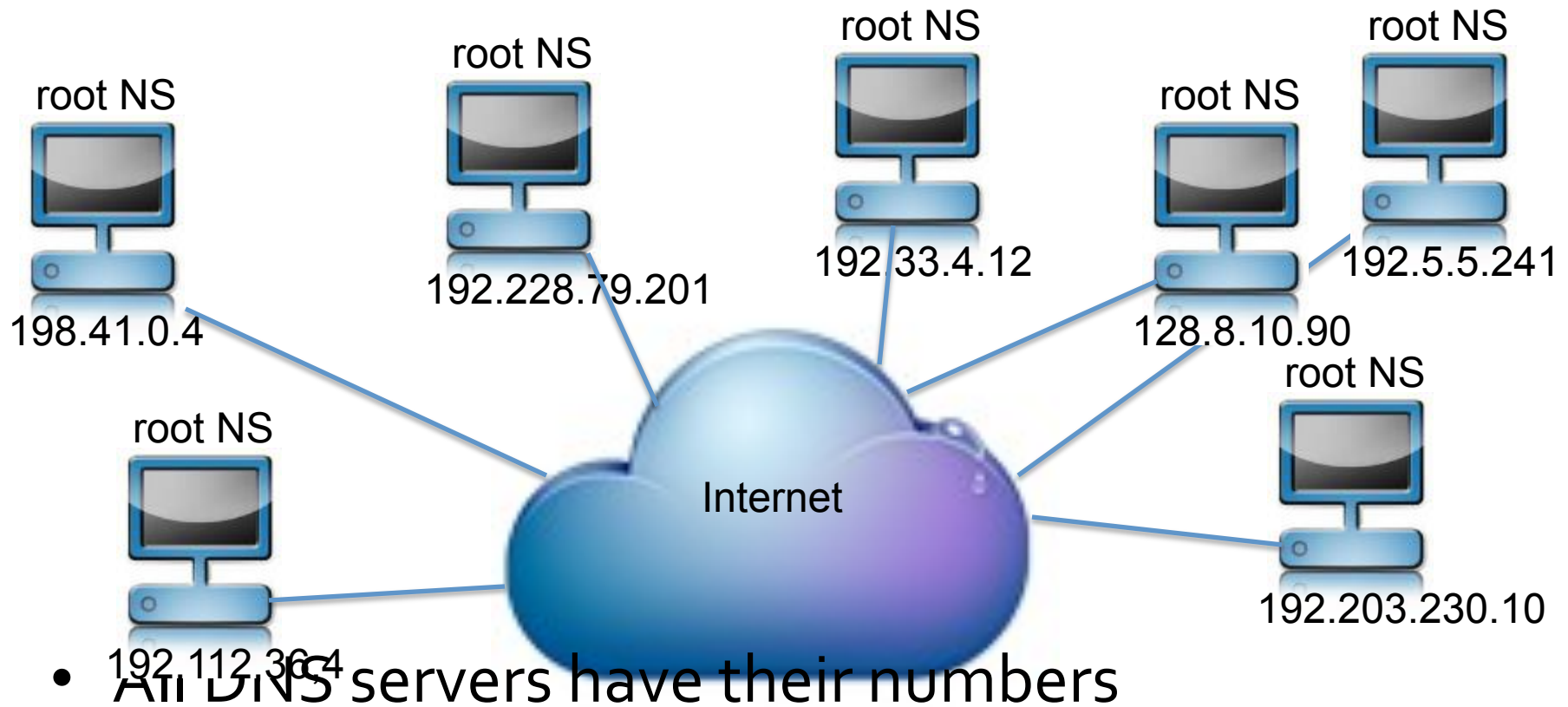


So, Here's How It Goes ...

- Your computer's DNS server has never heard of `cise.ufl.edu.root` ... so it pulls the domain name apart:
 - `cise`, a computer in the `.ufl` domain
 - `ufl`, a domain in the `.edu` domain
 - `edu`, a domain in the `.root` domain
- So, the DNS begins at the end and starts asking for the numbers of the autho computers ... who's the autho for the `.root` domain?

The DNS Design: Fact 2

- There are 13 Internet “root name servers” scattered around the world ... all the same



So, Here's How It Goes ...

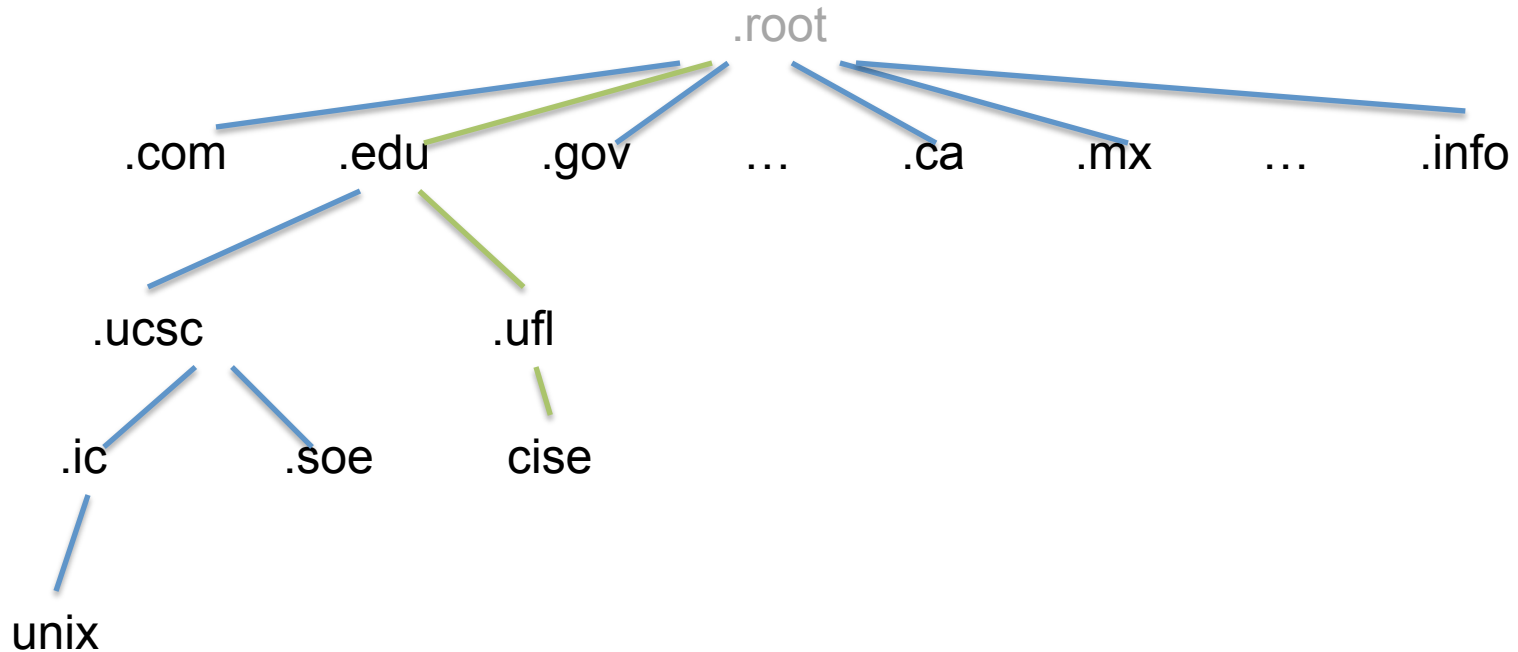
- Your computer's DNS server has never heard of `cise.ufl.edu.root` ... so it pulls the domain name apart:
 - `cise`, a computer in the `.ufl` domain
 - `ufl`, a domain in the `.edu` domain
 - `edu`, a domain in the `.root` domain
- So, the DNS begins at the end and starts asking for the numbers of the autho computers ... who's the autho for the `.root` domain?

Your DNS Asks the .root NS

- Please give me the number of **.edu** autho
 - Getting that it asks it, ...
- Please give me the number of **.ufl** autho
 - Getting that it asks it, ...
- Please give me the number of the **cise** machine
 - Getting **128.227.205.2**, it addresses your email and sends it on
- Simplification: it might have cached **.edu** autho and **.ufl** autho, which saves those requests

Logical Names Form Hierarchy

- As a hierarchy, it can be shown as a tree:



- The DNS is simply “walking” down the tree asking each autho for the number of next item

Suppose A Domain Adds Computer

- When a domain, say **.ufl**, adds a new computer it gets a name and an IP-address
- They add its name and number to the list in ufl autho' s memory and it's up and running, “known to the world”
- This is a completely decentralized solution – no one needs to be in charge except to make sure that the domain autho is up & correct

Properties ...

- **Fault tolerant:** when a hurricane takes out Miami's power, only the domains without power are affected ...
- **Robust:** when a fire burns down the building of a .root name server, 12 others can carry the load
- **Enormous capacity:** most lookups are independent and do not collide (b/c higher level domain authos are cached), but more capacity is possible by replicating authos

Compare DNS Structure To ..

- Master List Solution ...
 - Suppose the design was for the root NS computers to have a master list of all
domain_name: IP-address
pairs connected to the Internet
- How would it be different, better or worse?

HTML, CSS, JavaScript

Encoding Information: There's more!

- Bits and bytes encode the information, but that's not all
- **Tags** encode format and some structure in word processors
- **Tags** encode format and some structure in HTML
- **Tags** are one form of meta-data
- **Meta-data** is information about information



Windows Codepage 1251

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00	NUL	STX	SOT	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
01	000	001	002	003	004	005	006	007	008	009	010	011	012	013	014	015
10	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
016	017	018	019	020	021	022	023	024	025	026	027	028	029	030	031	
20	SP	!	#	\$	%	&	()	*	+	,	-	.	/		
032	033	034	035	036	037	038	039	040	041	042	043	044	045	046	047	
0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?	
30	048	049	050	051	052	053	054	055	056	057	058	059	060	061	062	063
@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	
40	064	065	066	067	068	069	070	071	072	073	074	075	076	077	078	079
P	Q	R	S	T	U	V	W	X	Y	Z	[]				
50	080	081	082	083	084	085	086	087	088	089	090	091	092	093	094	095
.	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	
60	096	097	098	099	100	101	102	103	104	105	106	107	108	109	110	111
p	q	r	s	t	u	v	w	x	y	z	{	}	--			
70	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
Ж	Г	,	г	"	...	†	Y	X	У	Ъ	←	Н	К	Г	И	
80	128	129	8218	402	8222	8230	8224	8225	710	8240	352	8249	338	141	142	143
ж	.	"	"	"	"	"	"	"	™	ь	»	ќ	ћ	и	о	
90	144	8216	8217	8220	8221	8226	8211	8212	732	8482	353	8250	339	157	158	376
Š	š	Ј	Ѡ	Ѓ	Ѕ	Є	©	©	«	»	–	®	Ї			
A0	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
°	±	І	і	г	о	Ѡ	·	ѳ	№	©	»	ј	ѕ	т		
B0	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	
C0	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207
Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	
D0	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223
а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п	
E0	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э	ю	я	
F0	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255

HTML and the Web

- The Web uses **http://** protocol
- Its asking for a Web page, which usually means a page expressed in **hyper-text markup language**, or HTML
 - *Hyper-text* refers to text containing [LINKS](#) that allow you to leave the linear stream of text, see something else, and return to the place you left
 - *Markup language* is a notation to describe how a published document is supposed to look: what kinds of fonts, text color, headings, images, etc.



Basics of HTML #1

- Rule 1: Content is given directly; anything that is not content is given inside of tags
- Rule 2: Tags made of < and > and used this way:

Attribute&Value

```
<p style="color:red">This is paragraph.</p>
```

Start

Content

End

Tag

Tag

It produces: **This is paragraph.**

- Rule 3: Tags must be paired or “self terminated”

Basics of HTML #2

- Rule 4: An HTML file has this structure:

```
<html>
```

```
  <head><title>Name of Page</title></head>
```

```
  <body>
```

Actual HTML page description goes here

```
  </body>
```

```
</html>
```

- Rule 5: Tags must be properly nested
- Rule 6: White space is mostly ignored
- Rule 7: Attributes (`style="color:red"`) preceded by space, name not quoted, value quoted

Basics of HTML #3

- To put in an image (.gif, .jpg, .png), use 1 tag

```

```

Tag Image Source
End

Alt Description

- To put in a link, use 2 tags

```
<a href="http://users.soe.ucsc.edu/~charlie">Charlie's page</a>
```

Hyper-text reference – the link Anchor

End

- More on HTML (including good tutorials) at
- <http://www.w3schools.com/html/default.asp>

Which does not apply to HTML?

- A. It is the language used for the web browser and the web server to communicate over the Internet.
- B. It is the language used to express how a document should be displayed.
- C. It is a language that allows for “documents” to be created that are not linear. (A book with chapters is linear – you normally read from start to end in order.)
- D. All of A-C apply to HTML.

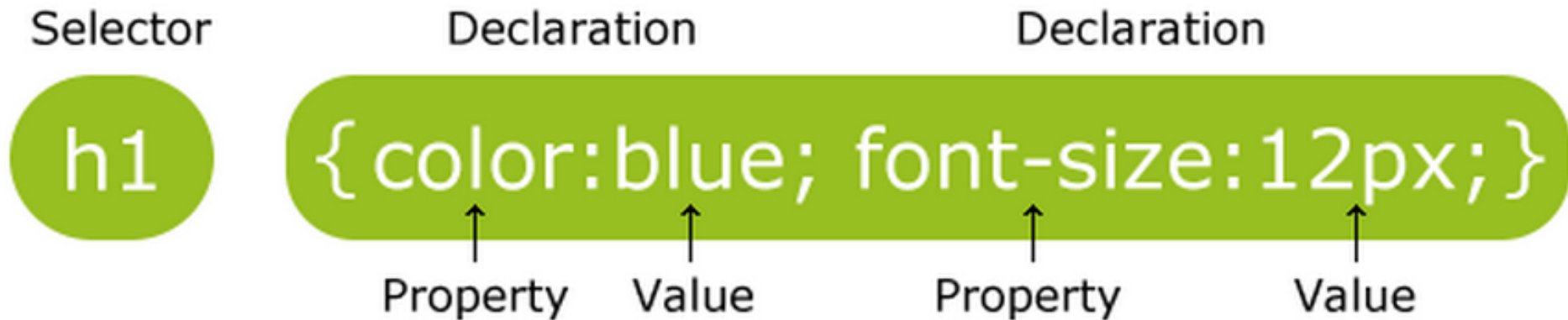
CSS – separating style from content

Content vs Style

- `<h1>This is a heading</h1>`
- `emphasize this`
- `Make this bold face`
- `<p style="color:red">This is paragraph is red.</p>`

CSS Syntax

A CSS rule has two main parts: a selector, and one or more declarations:



The selector is normally the HTML element you want to style.

Each declaration consists of a property and a value.

The property is the style attribute you want to change. Each property has a value.

From http://www.w3schools.com/css/css_syntax.asp

Adding CSS to your html file

```
<!DOCTYPE html>
<html><head>
<style>
p {color:red;text-align:center;}
body {background-image:url("images/
ComputerSlug.gif");}
</style>
</head>
<body>
<p>Hello World!</p>
<p>This paragraph is styled with CSS.</p>
</body></html>
```

Using an external CSS

```
<!DOCTYPE html>  
<html><head>  
<link rel="stylesheet" type="text/css"  
href="mystyle.css">  
</head>  
  
<body>  
<p>Hello World!</p>  
<p>This paragraph is styled with CSS.</p>  
</body></html>
```

mystyle.css

```
p {color:red;text-align:center;}
```

```
body {background-image:url("images/  
ComputerSlug.gif");}
```

Using inline style annotations

```
<!DOCTYPE html>
```

```
<html><head>
```

```
</head>
```

```
<body>
```

```
<p>Hello World!</p>
```

```
<p style="color:red;text-align:center;">This  
paragraph is styled with CSS.</p>
```

```
</body></html>
```


- A. css allows you to separate the specific formatting information from the main body of the document.
- B. css allows you to change how a lot of html documents will be displayed by changing just a single file
- C. A & B
- D. neither A nor B

JavaScript

Puts code right in the web page

- Syntax similar to Java
- Has its own set of predefined functions you need to discover (like Processing's drawing functions).
- How do we put it in the web page?

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<script>
```

```
function myFunction() {
```

```
  document.getElementById("demo").innerHTML="Goodbye";
```

```
}
```

```
</script>
```

```
</head>
```

```
<body>
```

```
<h1>My Web Page</h1>
```

```
<p id="demo">Hello!</p>
```

```
<button type="button" onclick="myFunction()">Click Me!</button>
```

```
</body>
```

```
</html>
```

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<h1>My Web Page</h1>
```

```
<p id="demo">Hello!</p>
```

```
<button type="button" onclick="myFunction()">Click Me!</button>
```

```
<p><strong>Note:</strong> myFunction is stored in an external file called  
"goodbye.js".</p>
```

```
<script src="goodbye.js"></script>
```

```
</body>
```

```
</html>
```

goodbye.js

```
function myFunction()  
{  
  document.getElementById("demo").innerHTML="Go  
odbye!";  
}
```

```
<!DOCTYPE html><html><body>
<!-- From http://www.w3schools.com/js/tryit.asp?filename=tryjs_ifthenelse -->
<p>Click the button to get a time-based greeting.</p>
<button onclick="myFunction()">Try it</button>
<p id="demo"></p>
<script>
function myFunction() {
  var x="";
  var time=new Date().getHours();
  if (time<20) {
    x="Good day";
  } else {
    x="Good evening";
  }
  document.getElementById("demo").innerHTML=x;
}
</script>

</body></html>
```

More Bits and Bytes

Huffman Coding

Encoding Text: How is it done?
ASCII, UTF, Huffman algorithm

ASCII

0100 0011
0100 0001
0101 0100

C
A
T



ASCII	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0000	N _U	S _H	S _X	E _X	E _T	E _O	A _K	B _L	B _S	H _T	L _F	Y _T	F _F	C _R	S _O	S _I
0001	D _L	D ₁	D ₂	D ₃	D ₄	N _K	S _V	E _Σ	C _N	E _M	S _B	E _C	F _S	G _S	R _S	U _S
0010		!	"	#	\$	%	&	'	()	*	+	,	-	.	/
0011	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
0100	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
0101	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
0110	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
0111	p	q	r	s	t	u	v	w	x	y	z	{		}	~	D _T
1000	S ₀	S ₁	S ₂	S ₃	I _N	N _L	S _S	E _S	H _S	H _J	Y _S	P _D	P _V	R _I	S ₂	S ₃
1001	D _C	P ₁	P ₂	S _E	C _C	M _M	S _P	E _P	O _B	O _O	O _A	C _S	S _T	O _S	P _M	A _P
1010	A _o	ı	ç	£	♀	¥	ı	§	¨	©	♂	«	¬	-	®	—
1011	°	±	²	³	´	μ	¶	·	¸	¹	º	»	¼	½	¾	¿
1100	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
1101	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
1110	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
1111																

0100 0111 | 0110 1111 | 0010 0000 | 0101 0011 | 0110 1100 | 0111 0101 | 0110 0111 | 0111 0011 |

UTF-8: All the alphabets in the world

- Uniform Transformation Format: a variable-width encoding that can represent every character in the Unicode Character set
- 1,112,064 of them!!!
- <http://en.wikipedia.org/wiki/UTF-8>
- UTF-8 is the dominant character encoding for the World-Wide Web, accounting for more than half of all Web pages.
- The Internet Engineering Task Force (IETF) requires all Internet protocols to identify the encoding used for character data
- The supported character encodings must include UTF-8.

لماذا لا يتكلمون اللغة العربية فحسب؟
Зачо те просто не могат да говорят **български**?
Per què no poden simplement parlar en **català**?
他們爲什麼不說中文(台灣)?
Proč prostě nemluví **česky**?
Hvorfor kan de ikke bare tale **dansk**?
Warum sprechen sie nicht einfach **Deutsch**?
Μα γιατί δεν μπορούν να μιλήσουν **Ελληνικά**?
Why can't they just speak English?
¿Por qué no pueden simplemente hablar en **castellano**?
Miksi he eivät yksinkertaisesti puhu **suomea**?
Pourquoi, tout simplement, ne parlent-ils pas **français**?
למה הם פשוט לא מדברים **עברית**?
Miért nem beszélnek egyszerűen **magyarul**?
Af hverju geta þeir ekki bara talað **íslensku**?
Perché non possono semplicemente parlare **italiano**?
なぜ、みんな日本語を話してくれないのか?
세계의 모든 사람들이 한국어를 이해한다면 얼마나 좋을까?
Waarom spreken ze niet gewoon **Nederlands**?
Hvorfor kan de ikke bare snakke **norsk**?
Dlaczego oni po prostu nie mówią po **polsku**?
Porque é que eles não falam em **Português (do Brasil)**?
Oare ăștia de ce nu vorbesc **românește**?
Почему же они не говорят **но-русски**?
Zašto jednostavno ne govore **hrvatski**?
Pse nuk duan të flasin vetëm **shqip**?
Varför pratar dom inte bara **svenska**?
ทำไมเขาถึงไม่พูดภาษาไทย
Neden **Türkçe** konuşamıyorlar?

UTF is a VARIABLE LENGTH ALPHABET CODING

- Remember ASCII can only represent 128 characters (7 bits)
- UTF encodes over one million
- Why would you want a variable length coding scheme?

Bits	Last code point	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
7	U+007F	0xxxxxxx					
11	U+07FF	110xxxxx	10xxxxxx				
16	U+FFFF	1110xxxx	10xxxxxx	10xxxxxx			
21	U+1FFFFFF	11110xxx	10xxxxxx	10xxxxxx	10xxxxxx		
26	U+3FFFFFFF	111110xx	10xxxxxx	10xxxxxx	10xxxxxx	10xxxxxx	
31	U+7FFFFFFF	1111110x	10xxxxxx	10xxxxxx	10xxxxxx	10xxxxxx	10xxxxxx

UTF-8

Bits of code point	First code point	Last code point	Bytes in sequence	Byte 1	Byte 2	Byte 3	Byte 4
7	U+0000	U+007F	1	0xxxxxxx			
11	U+0080	U+07FF	2	110xxxxx	10xxxxxx		
16	U+0800	U+FFFF	3	1110xxxx	10xxxxxx	10xxxxxx	
21	U+10000	U+1FFFFFF	4	11110xxx	10xxxxxx	10xxxxxx	10xxxxxx

UTF-8

Bits of code point	First code point	Last code point	Bytes in sequence	Byte 1	Byte 2	Byte 3	Byte 4
7	U+0000	U+007F	1	0xxxxxxx			
11	U+0080	U+07FF	2	110xxxxx	10xxxxxx		
16	U+0800	U+FFFF	3	1110xxxx	10xxxxxx	10xxxxxx	
21	U+10000	U+1FFFFFF	4	11110xxx	10xxxxxx	10xxxxxx	10xxxxxx

What is the first Unicode value represented by this sequence?

11101010 1000011 10000111 00111111 11000011 10000000

- A. 0000000001101010
- B. 0000000011101010
- C. 0000001010000111
- D. 1010000011000111

UTF-8

Bits of code point	First code point	Last code point	Bytes in sequence	Byte 1	Byte 2	Byte 3	Byte 4
7	U+0000	U+007F	1	0xxxxxxx			
11	U+0080	U+07FF	2	110xxxxx	10xxxxxx		
16	U+0800	U+FFFF	3	1110xxxx	10xxxxxx	10xxxxxx	
21	U+10000	U+1FFFFFF	4	11110xxx	10xxxxxx	10xxxxxx	10xxxxxx

How many Unicode characters are represented by this sequence?

11101010 1000011 10000111 00111111 11000011 10000000

- A. 1
- B. 2
- C. 3
- D. 4
- E. 5

How many bits for all of Unicode?

There are 1,112,064 different Unicode characters. If a fixed bit format (like ascii with its 7 bits) were used, how many bits would you need for each character? (Hint: $2^{10} = 1024$)

- A. 10
- B. 17
- C. 21
- D. 32
- E. 40

Coding can be used to do Compression

- What is CODING?
 - The conversion of one representation into another

- What is COMPRESSION?
 - Change the representation (digitization) in order to **reduce** size of data (number of bits needed to represent data)

- Benefits

- **Reduce storage** needed
 - Consider growth of digitized data.
- **Reduce** transmission cost / **latency** / bandwidth
- When you have a 56K dialup modem, every savings in BITS counts, **SPEED**
 - Also consider telephone lines, texting



Huffman Code Example

- “dog cat cat bird bird bird bird fish”

Symbol	Dog	Cat	Bird	Fish
Frequency	1/8	1/4	1/2	1/8
Original Encoding	00	01	10	11
	2 bits	2 bits	2 bits	2 bits
Huffman Encoding	110	10	0	111
	3 bits	2 bits	1 bit	3 bits

- Expected size

- Original $\Rightarrow 1/8 \times 2 + 1/4 \times 2 + 1/2 \times 2 + 1/8 \times 2 = 2$ bits / symbol
- Huffman $\Rightarrow 1/8 \times 3 + 1/4 \times 2 + 1/2 \times 1 + 1/8 \times 3 = 1.75$ bits / symbol

Huffman Code Example

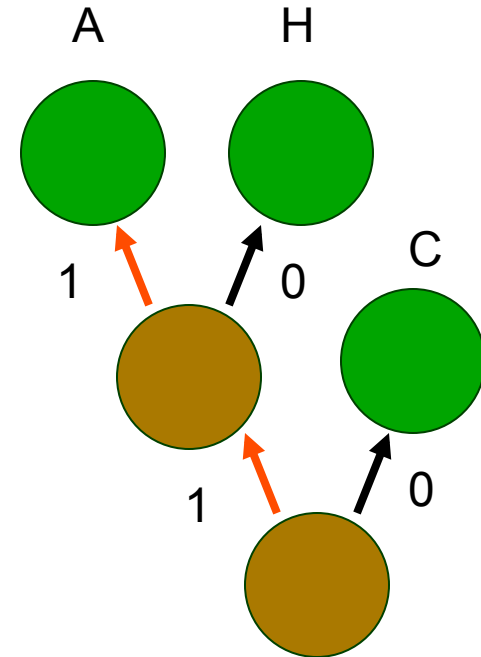
Symbol	Dog	Cat	Bird	Fish
Frequency	1/8	1/4	1/2	1/8
Original Encoding	00	01	10	11
	2 bits	2 bits	2 bits	2 bits
Huffman Encoding	110	10	0	111
	3 bits	2 bits	1 bit	3 bits

How many bits are saved using the above Huffman coding for the sequence Dog Cat Bird Bird Bird?

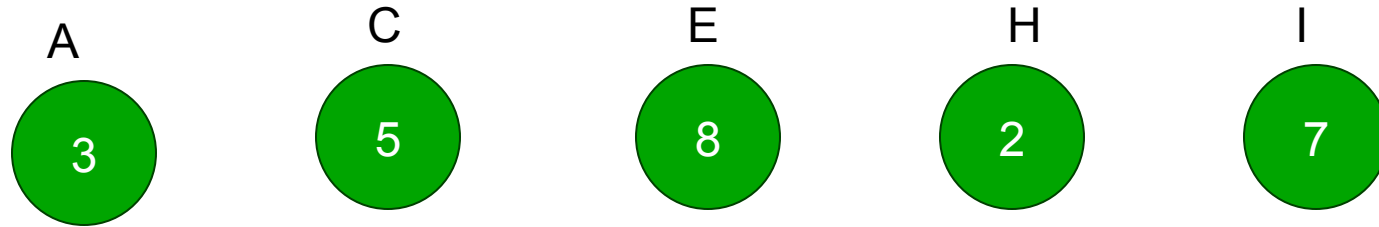
- A. 0 B. 1 C. 2 D. 3 E. 4

Huffman Code Algorithm: Data Structures

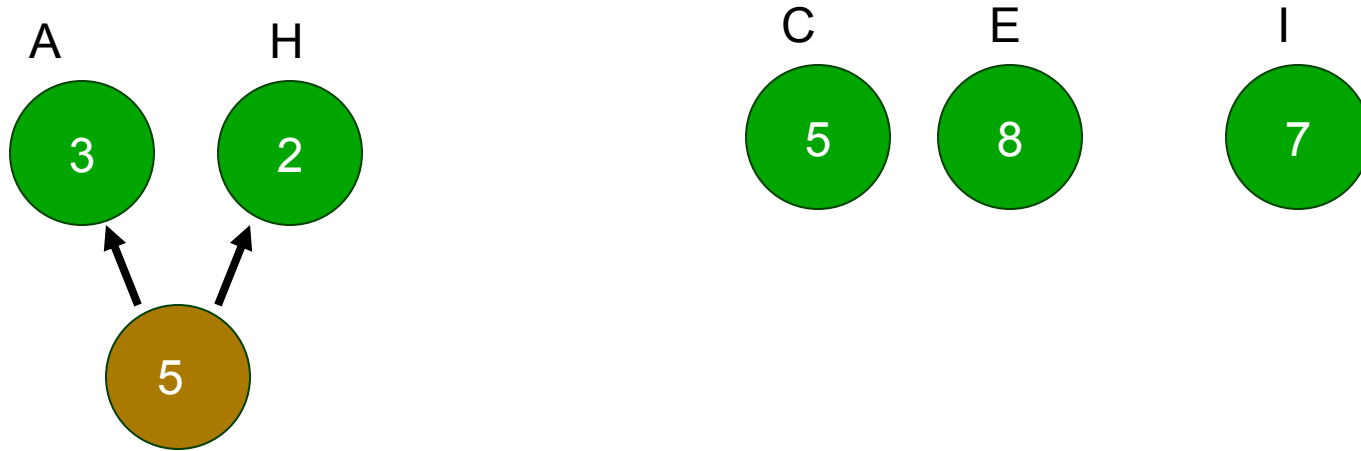
- Binary (Huffman) tree
 - Represents Huffman code
 - Edge \Rightarrow code (0 or 1)
 - Leaf \Rightarrow symbol
 - Path to leaf \Rightarrow encoding
 - Example
 - A = "11", H = "10", C = "0"
 - Good when ???
 - A, H less frequent than C in messages
- Want to efficiently build a binary tree



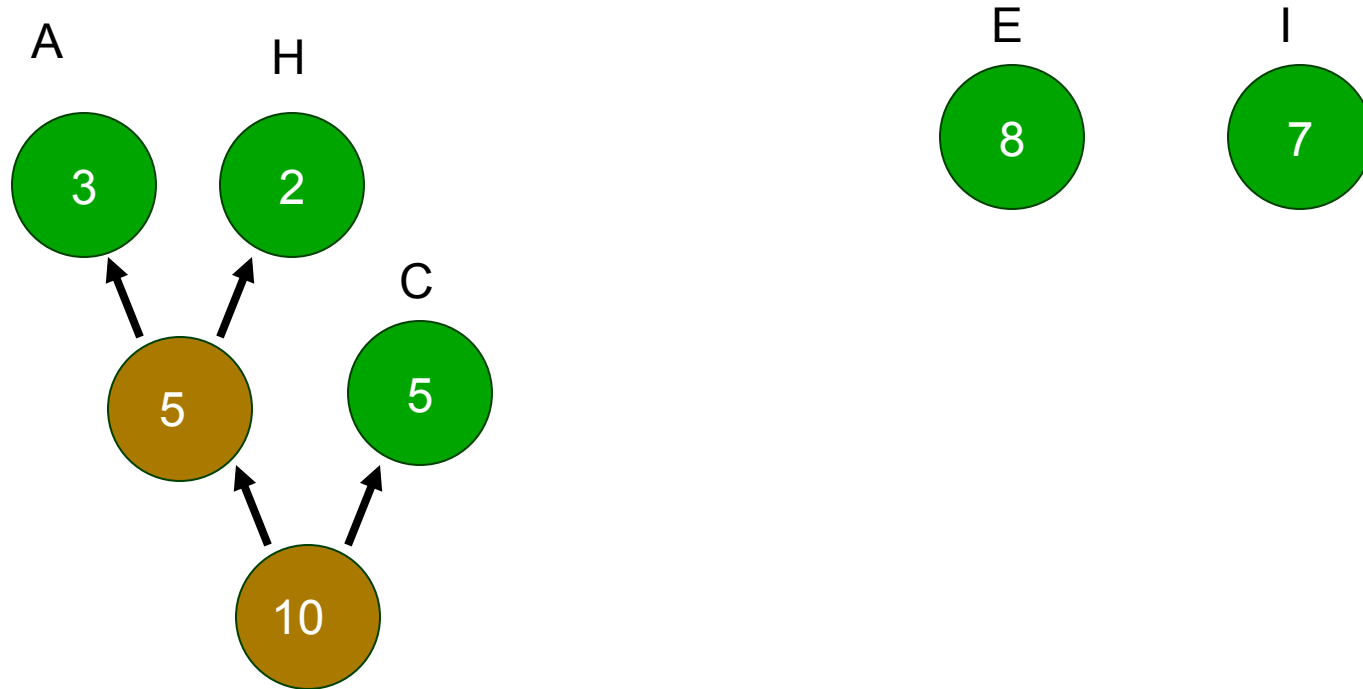
Huffman Tree Construction I



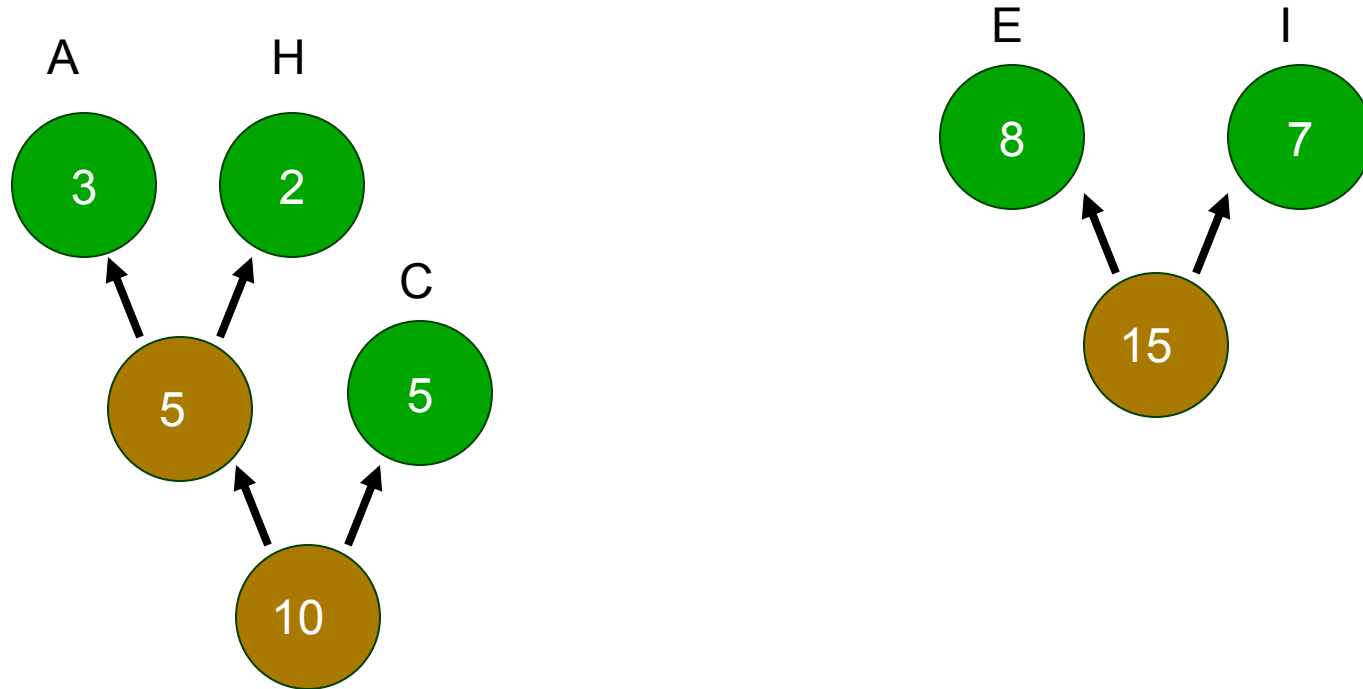
Huffman Tree Step 2: can first re-order by frequency



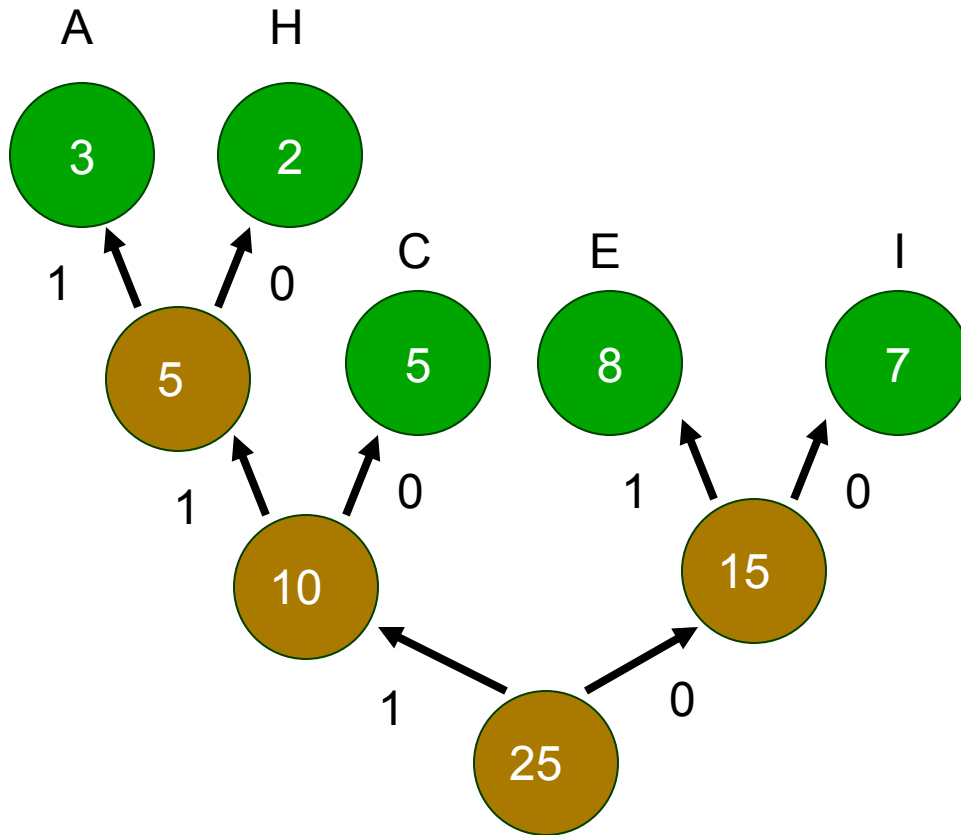
Huffman Tree Construction 3



Huffman Tree Construction 4



Huffman Tree Construction 5



E = 01

I = 00

C = 10

A = 111

H = 110

Huffman Coding Example

- Huffman code

E = 01

I = 00

C = 10

A = 111

H = 110

- Input

– ACE

- Output

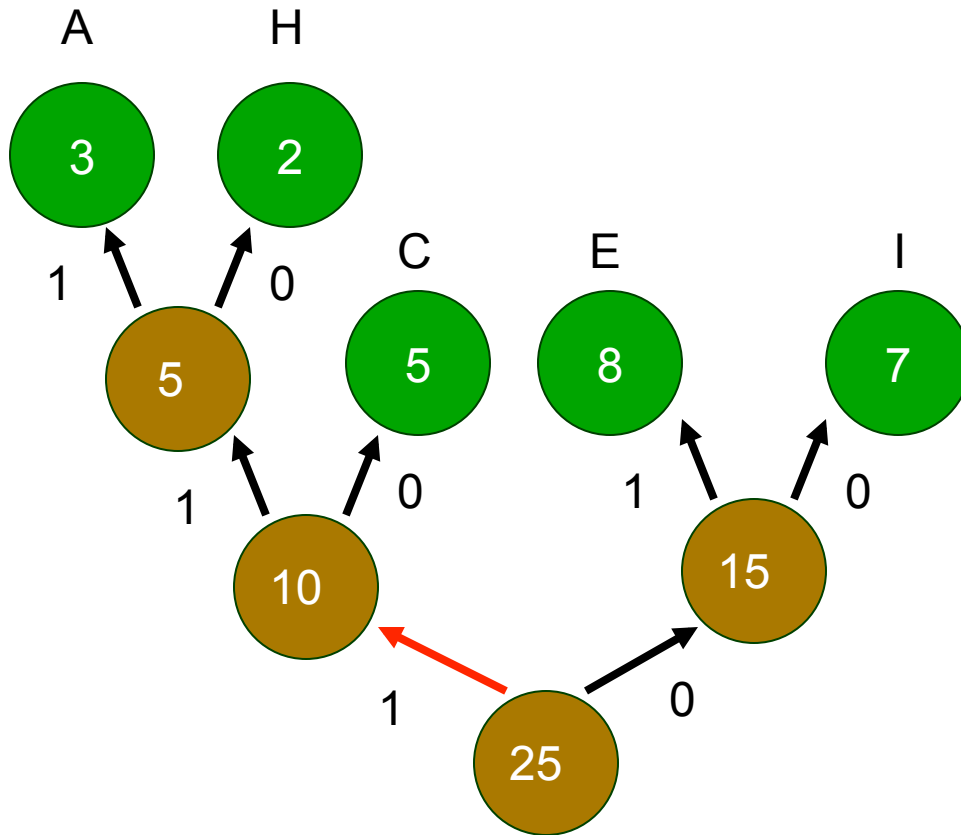
– (111)(10)(01) = 1111001

Huffman Code Algorithm Overview

- Decoding
 - Read compressed file & binary tree
 - Use binary tree to decode file
 - Follow path from root to leaf

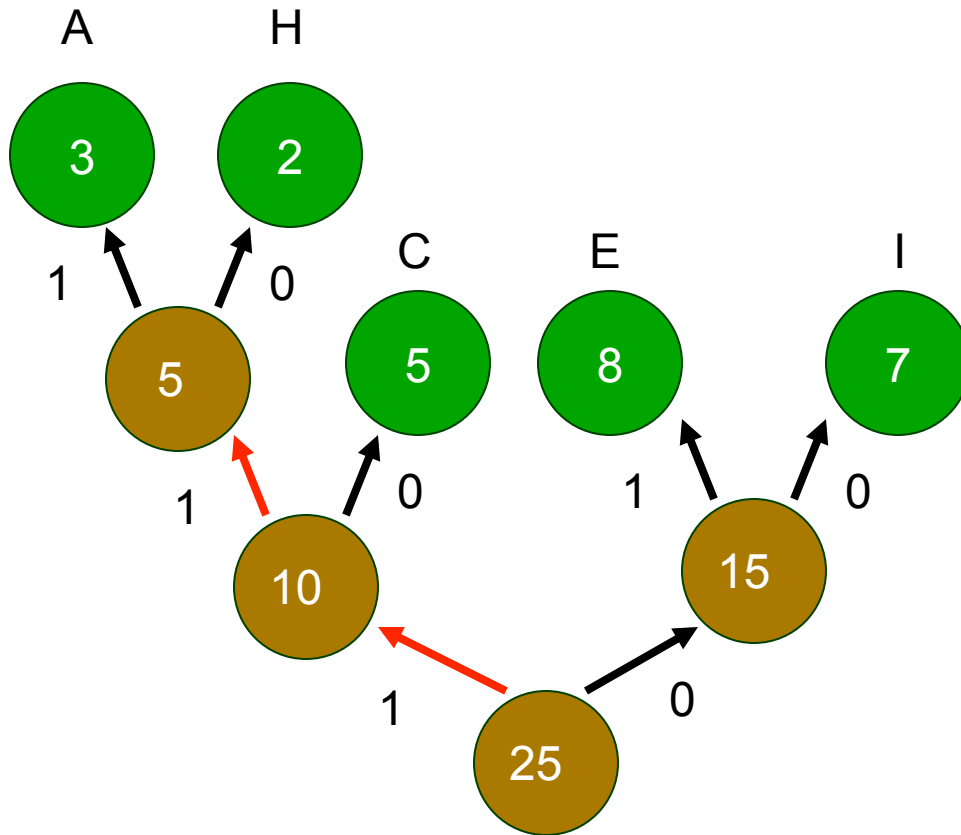
Huffman Decoding I

1111001

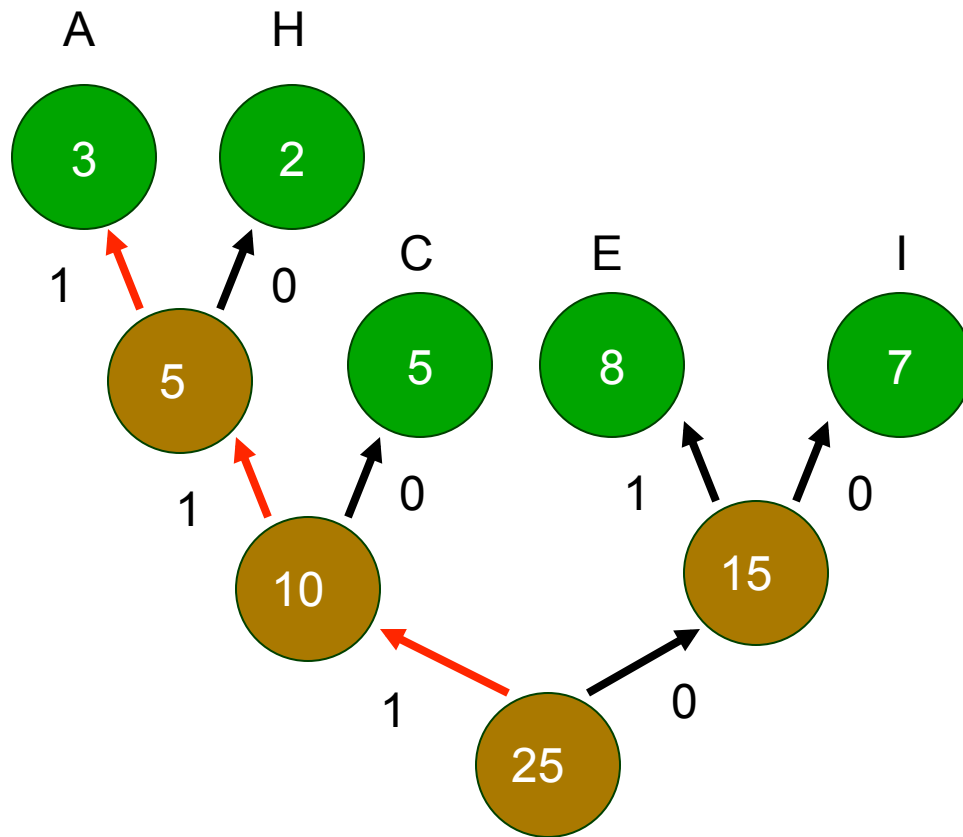


Huffman Decoding 2

1111001



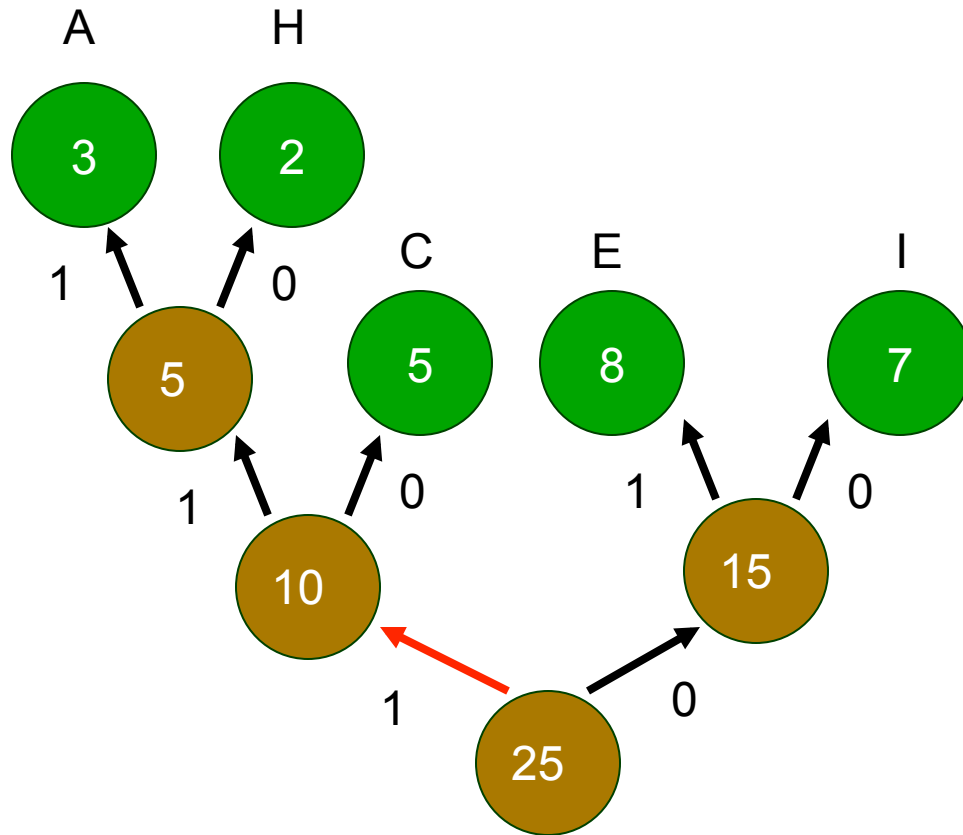
Huffman Decoding 3



1111001

A

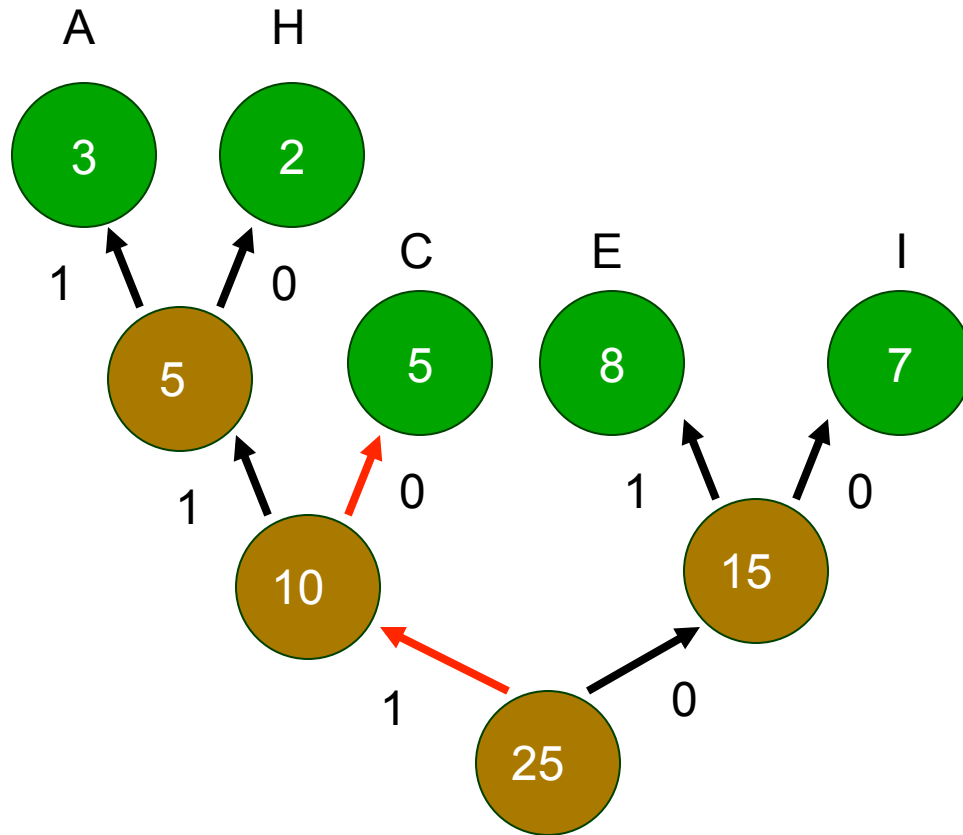
Huffman Decoding 4



1111001

A

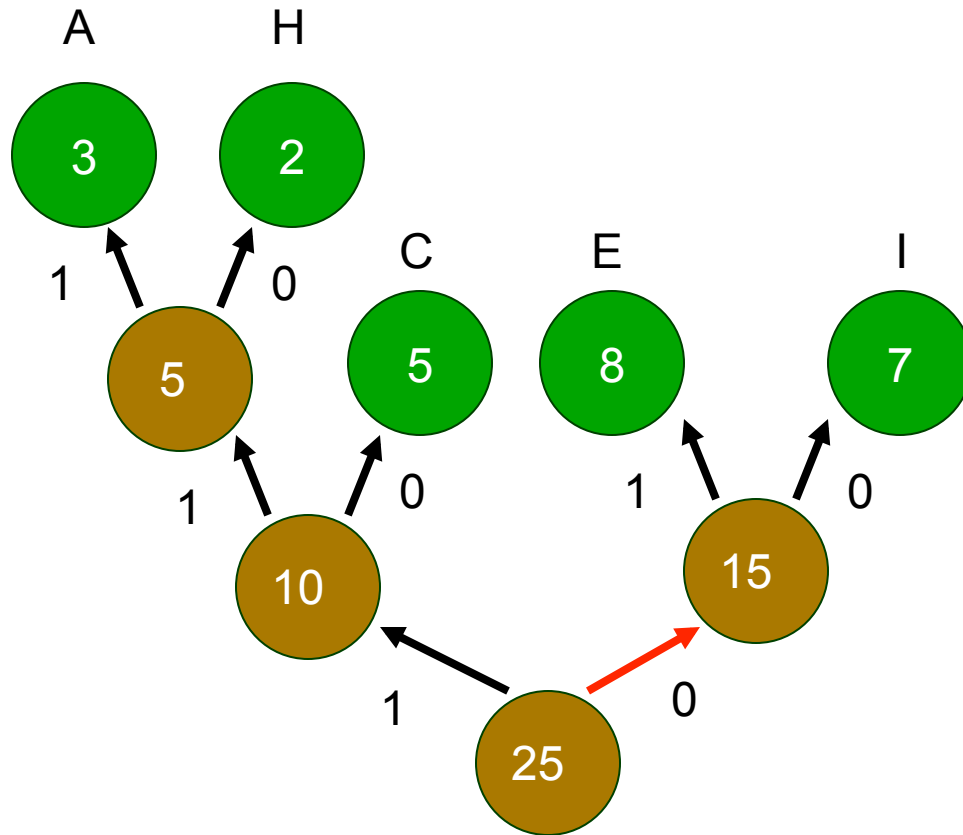
Huffman Decoding 5



1111001

AC

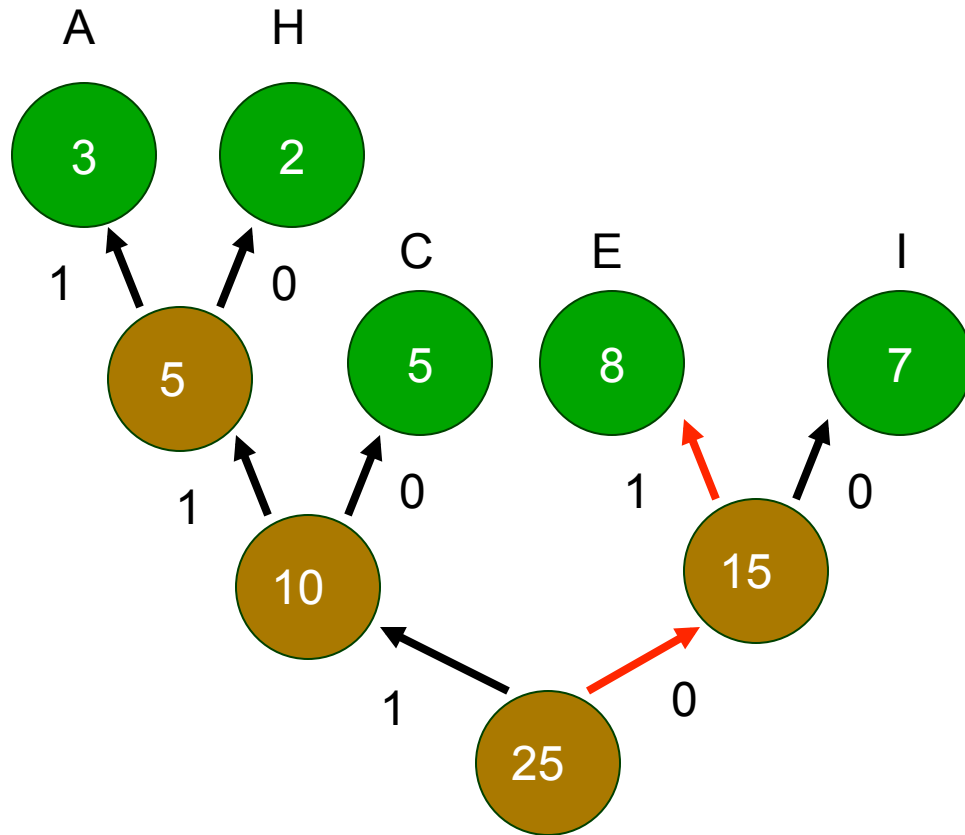
Huffman Decoding 6



1111001

AC

Huffman Decoding 7



1111001

ACE

What can a computer be commanded to do?

Sorting Complexity

*Based on slides from
Lawrence Snyder
University of Washington, Seattle*

Searching

- Guess a number between 1 and 100. How many guesses do you need?
- Scatter a deck of cards on the floor. How many do you have to turn over to find the ace of spades?

Sorting (exchange)

- Putting a sequence of items into alphabetical or numerical order

walrus seal whale gull clam

Algorithm: compare
to all following items,
reorder if needed

- Other ways to sort

wa se wh g c

se wa wh g c

g wa wh se c

c wa wh se g

c wa wh se g

c se wh wa g

c g wh wa se

c g wa wh se

c g se wh wa

c g se wa wh

Sorting (bubble)

- Putting a sequence of items into alphabetical or numerical order

walrus seal whale gull clam

Algorithm: compare adjacent items, reorder if needed

Bubble sort has n^2 Time

wa se wh g c
se wa wh g c
se wa wh g c
se wa g wh c
se wa g c wh
se wa g c wh
se wa g c wh
se g wa c wh
se g c wa wh
se g c wa wh
g se c wa wh
g c se wa wh
g c se wa wh
c g se wa wh

How Long To Sort w/Exchange?

- The pattern is, for n items

n-1 focus on first time

n-2 focus on second item

n-3 focus on third item

...

1 on next to last

n-1 rows in list, avg of each

row is $n/2$, so $(n-1) \times n/2$

$$= (n^2 - n)/2$$

- Time proportional to n^2

wa se wh g c

se wa wh g c

g wa wh se c

c wa wh se g

c wa wh se g

c se wh wa g

c g wh wa se

c g wa wh se

c g se wh wa

c g se wa wh

There are Different Algorithms

- Is there a better way to do sorting?
- **QUICKSORT**
 - **Fastest** known sorting algorithm in practice
 - Average case: $O(N \log N)$ (we don't prove it)
 - Worst case: $O(N^2)$
 - But, the worst case seldom happens.
 - A divide-and-conquer recursive algorithm
- Video of selection vs quicksort: http://youtu.be/cVMKXKoGu_Y

Polynomial

- Other computations have running time
 - proportional to n^3 – matrix multiplication
 - proportional to n^4
 - ...
- All of them are lumped together as “*polynomial* time computations”
 - Considered to be realistic ... a person can wait
 - Polynomial, but not linear ... get a computer person to help develop your solution

Which expression grows slowest as N gets bigger?

A. Time proportional to N^2

B. Time proportional to $N \cdot \log(N)$

To Infinity And Beyond

There are more complex computations ...

Suppose you want to visit 28 cities in the US (for a rock concert?) and you want to minimize how much you pay for airplane tickets

You could select an ordering of cities (SEA → PDX → SFO → LAX ...) and compute the ticket price.

Then pick another ordering (SEA → SFO → LAX → PDX ...), compute this ticket price and compare to the previous one

Always keep the cheapest itinerary

This seems very dumb ... isn't there a better way?

Traveling Salesman Problem

- Actually, no one knows a way to solve this problem significantly faster than checking all routes and picking the cheapest ...
- Not polynomial time ... guessing, No Poly sol' n
- This is an NP-Complete problem
 - Many many related problems ... the best solution is “generate and check”
 - Best way to pack a container ship
 - Most efficient scheduling for high school students' classes
 - Least fuel to deliver UPS packages in Washington
 - Fewest public alert broadcast stations for US

Astonishing Fact

- Although there are thousands of NP-**Hard** Problems, meaning they're basically “generate and check” ...
 - NP-**Complete** computations (like traveling salesman) have the property that if any one of them can be done fast (n^x -time, say) then EVERYONE of the related problems can be too!
 - Is Traveling Salesman solvable in n^x time is one of the great open questions in computer science

Be Famous ... Answer This Question

Finding an item in an unsorted list

What is the time complexity of finding an item in an unsorted list?

- A. Less than linear ($\log(n)$ or constant)
- B. Linear
- C. Polynomial but more time than linear
- D. NP (exponential like 2^n)

There's Stuff A Computer Can't Do

- Some problems are too big – combinatorial explosion – like checking each chess game to see if there is a guaranteed win for White
 - Too many items to check
 - Doable in principle, however

Some Well Formed Problems

- One problem that has a clear specification but can't be solved is

Halting Problem

Decide, given a program P and input Q whether $P(Q)$, that is, P run on input Q , will eventually stop running and give an answer

- This seems pretty easy ... though running it won't work because it might not stop ... but maybe analysis could find any errors

NOT

- The halting problem cannot be solved
- “This statement is false.” True or false?
- Here’s why ...
 - Suppose (for purposes of contradiction) that some program $H(P, Q)$ answers the halting problem (will it halt and give an answer) for program P on data Q
 - Notice the question is not, does it give the RIGHT answer ... just will it give any answer

Finding an item in a sorted list

What is the time complexity of finding an item in a sorted list?

- A. Less than linear ($\log(n)$ or constant)
- B. Linear
- C. Polynomial but more time than linear
- D. NP (exponential like 2^n)
- E. Unsolvability (like the halting problem)

Good Luck on the Final!