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

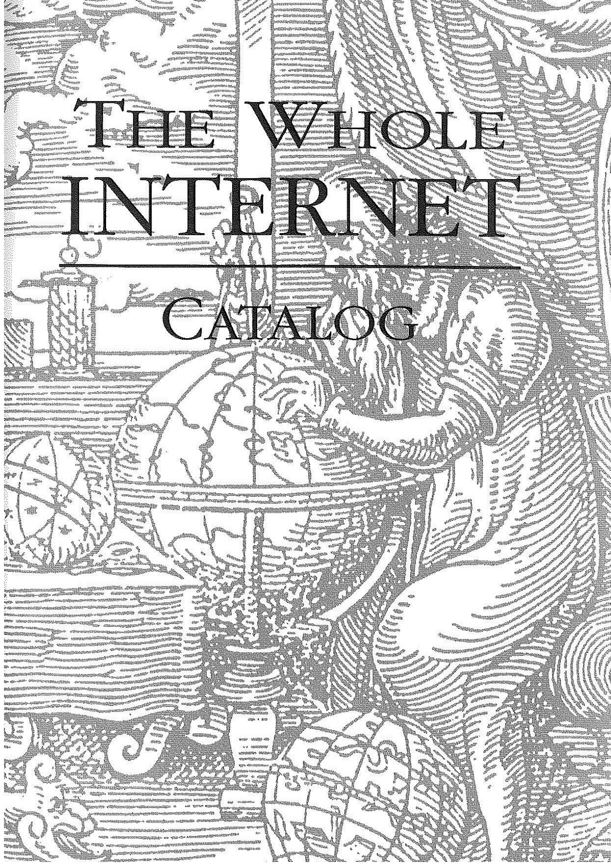
- search as censor
- right to be forgotten
- can’t take it back
- free speech
- regulate search engines?

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 seach 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.

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

ZWKBJLSIHCFGRAXEMNTUQPDOVY

- 26! or about $4\text{x}10^{26}$ 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

HIJKLMNOPQRSTUVWXYZABCDEFG

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

A. 3

B. 4

C. 26 X 3 = 78

D. 26 X 4 = 104

Vigenère Cipher

ABCDEFGHIJKLMNOPQRSTUVWXYZ

NOPQRSTUVWXYZABCDEFGHIJKLM

DEFGHIJKLMNOPQRSTUVWXYZABC

HIJKLMNOPQRSTUVWXYZABCDEFG

- 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

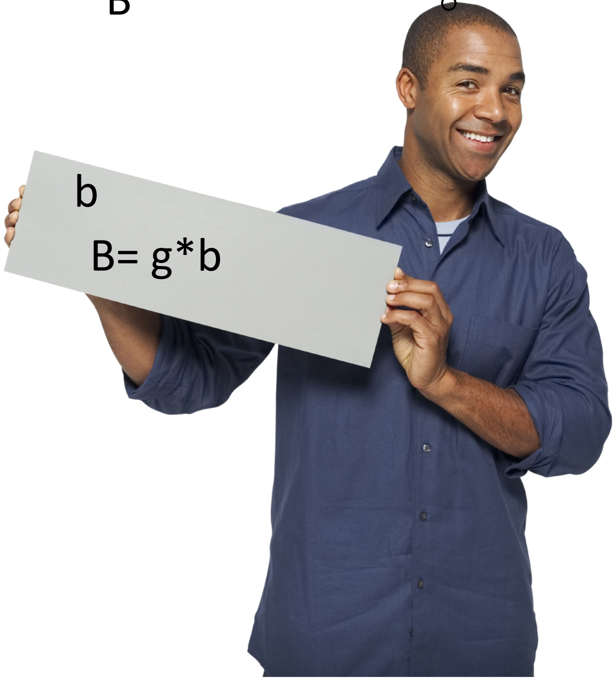
K=b*A=b*g*a

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

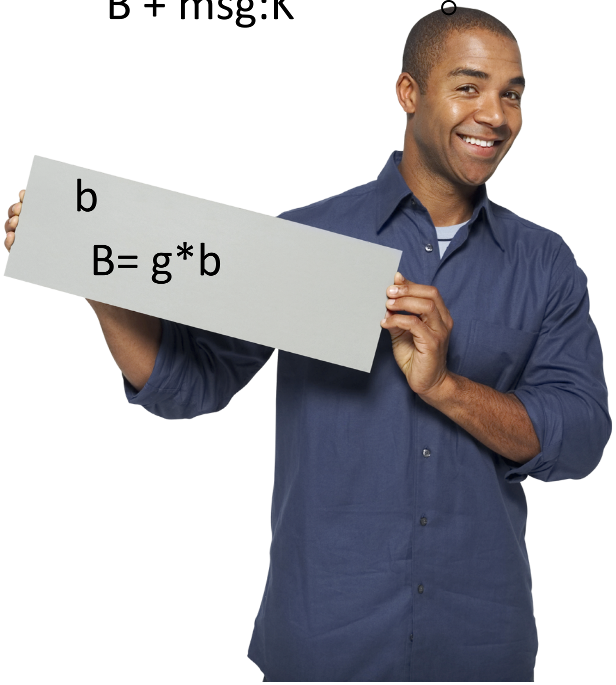
A

B + msg:K

Public Directory

a

A= g*a



b

B= g*b

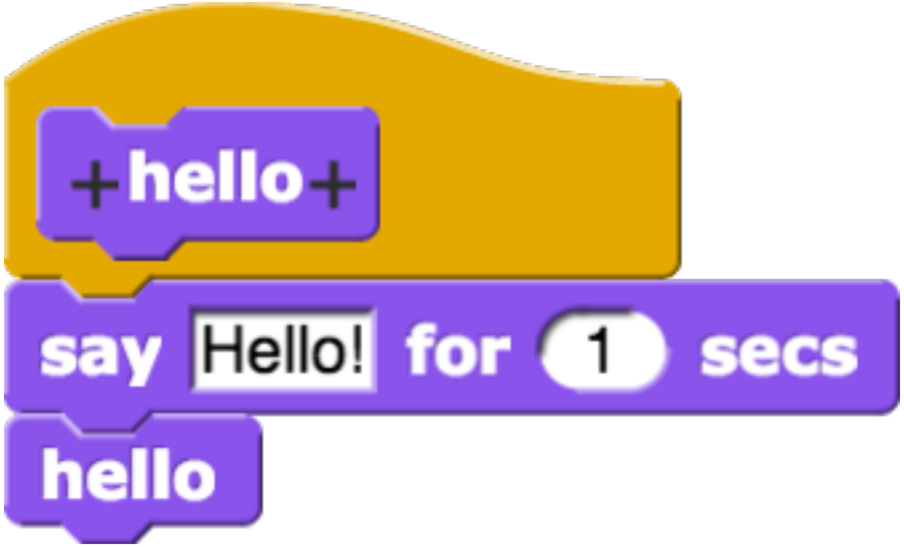
A Shared Secret?

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.

Snap Practice

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



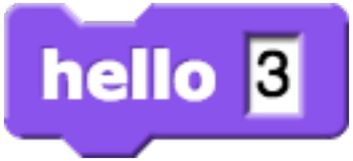
```
+hello+
say Hello! for 1 secs
hello
```

A. 0

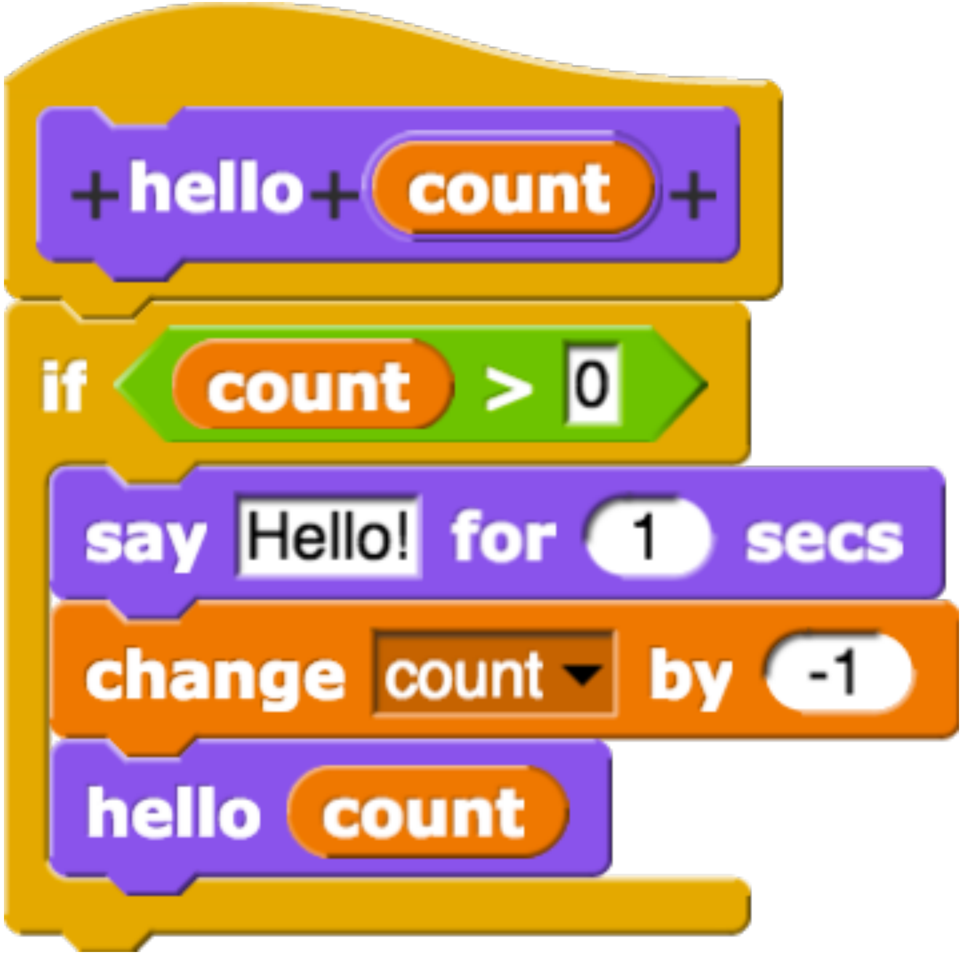
B. 1

C. forever

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



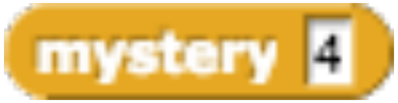
```
hello 3
```



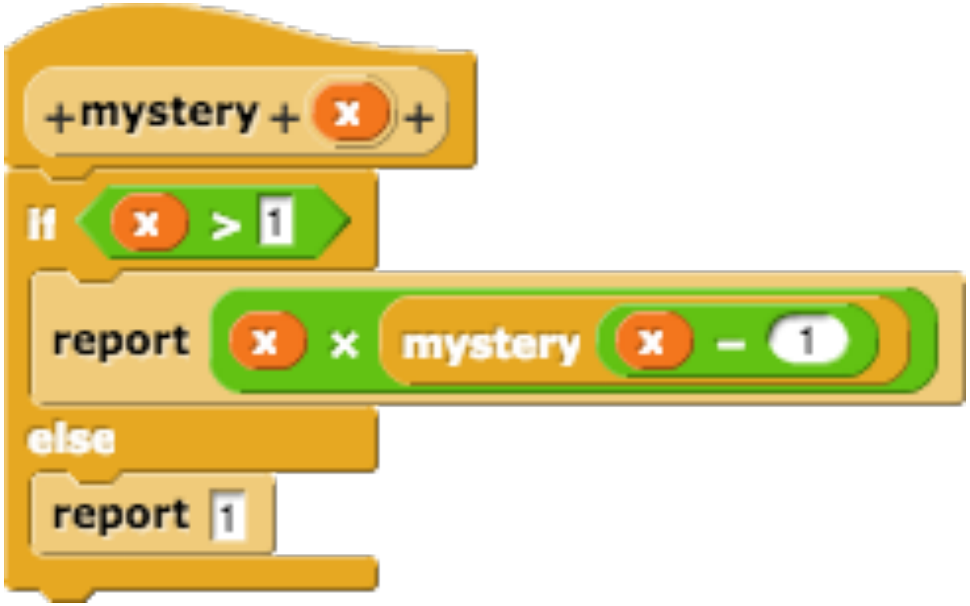
```
+hello+(count)+
if <(count) > 0>
    say Hello! for 1 secs
    change count by -1
    hello (count)
```

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

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!

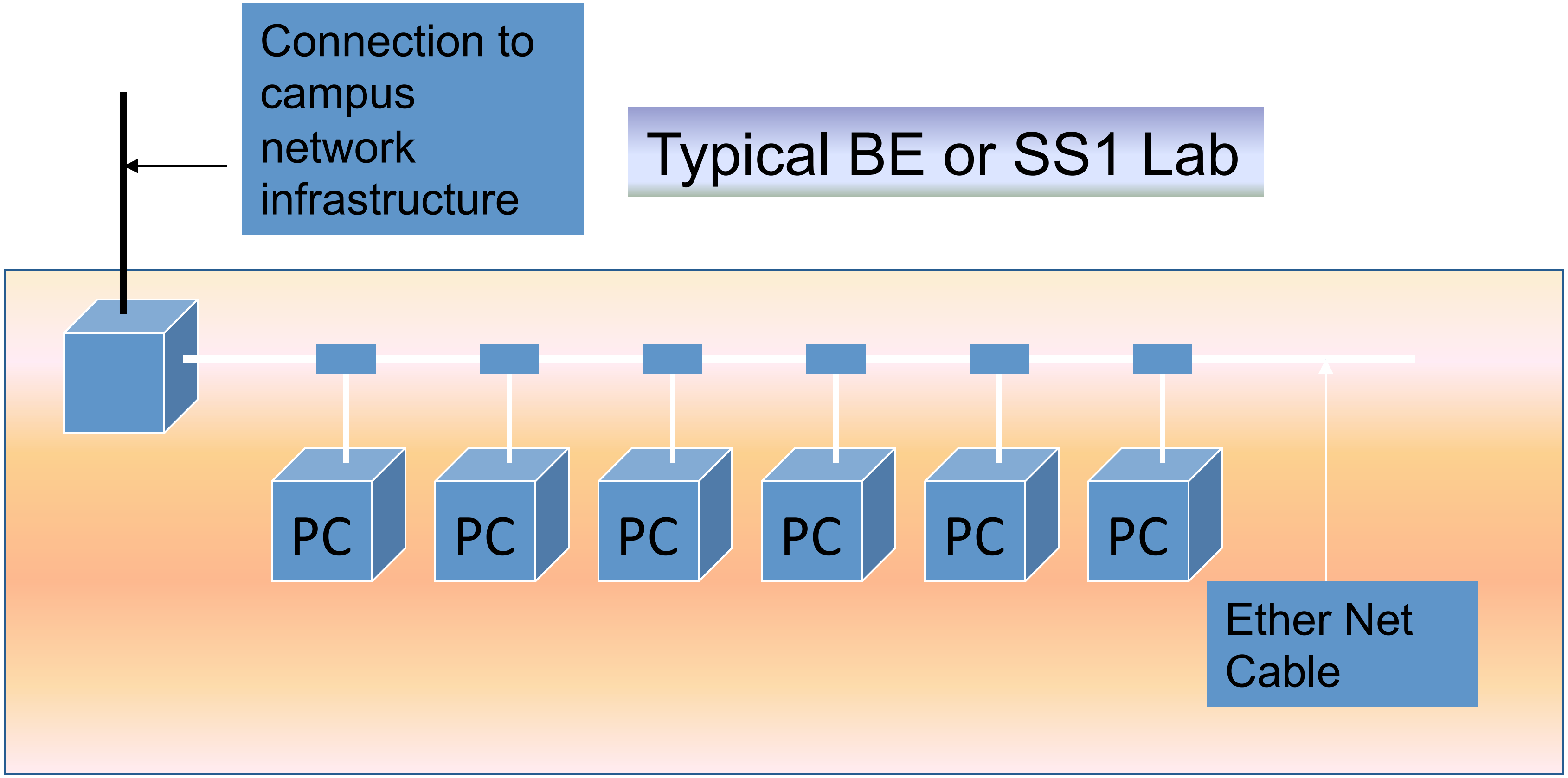
To communicate computers need to know how to set up the info to be sent and interpret the info received

- 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



Connection to campus network infrastructure

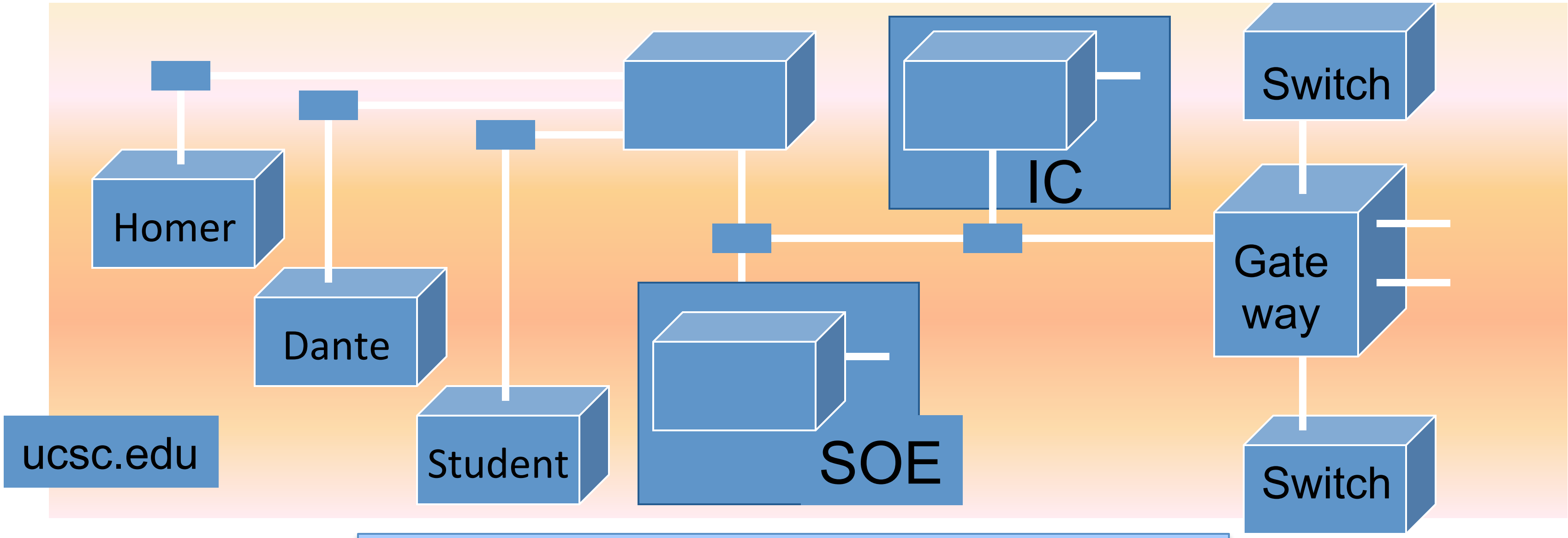
Typical BE or SS1 Lab

Ether Net Cable

© 2010 Larry Snyder, CSE

Campus & The World

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



All communication by TCP/IP

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

addr # data

- Each makes its way separately to destination, possibly taking different routes
- Reassembled at destination forming msg

Key Point: Taking separate routes lets packets by-pass 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.

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

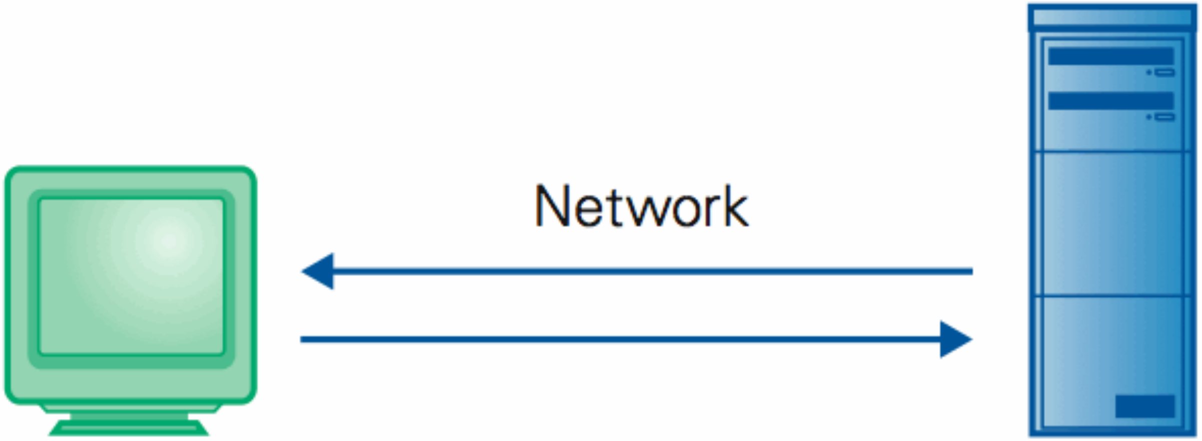
- Many people mis-use the terms “Internet” and “World Wide Web”
- Let’ s get them right

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



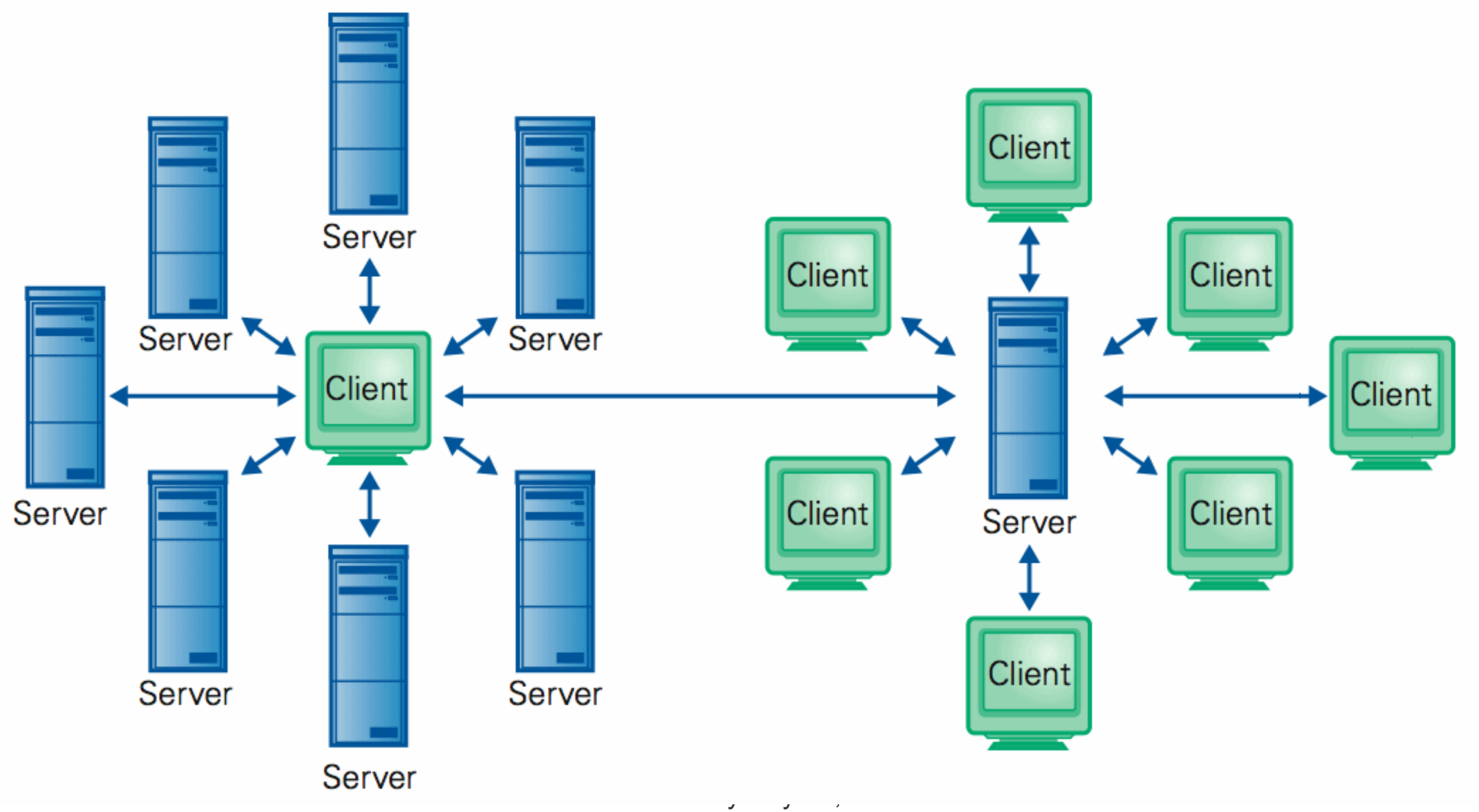
Client Computer
Requests services
(Sends URL for a Web page)

Server Computer
Provides services
(Returns the Web page file)

- It’s a VERY BRIEF relationship

Client/Server Is Also Smart

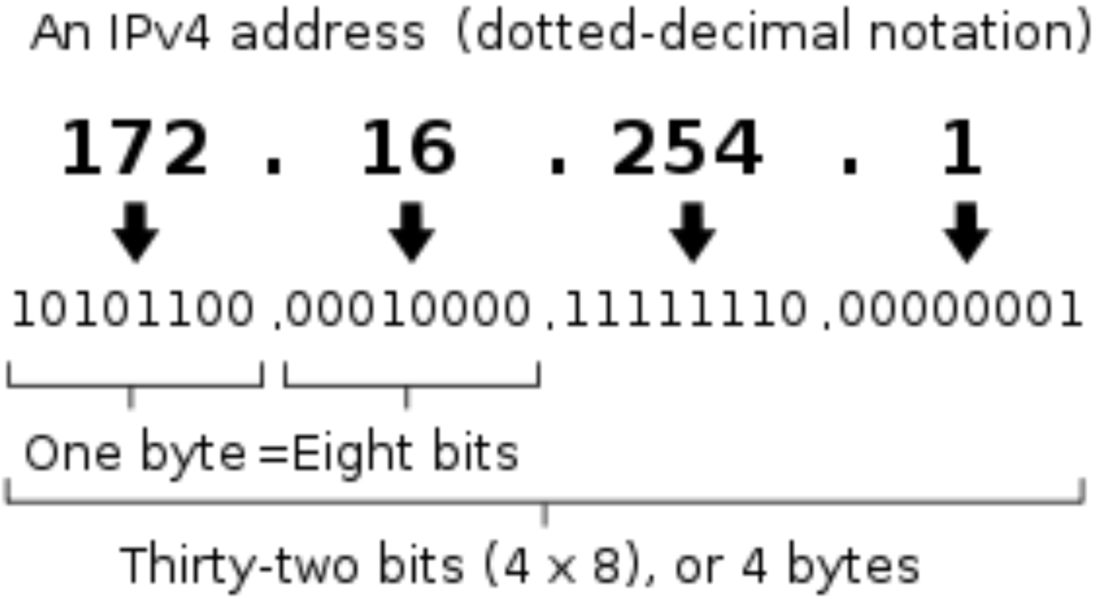
- Clients and servers are not connected – they only exchange info … "no commitment issues"



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.



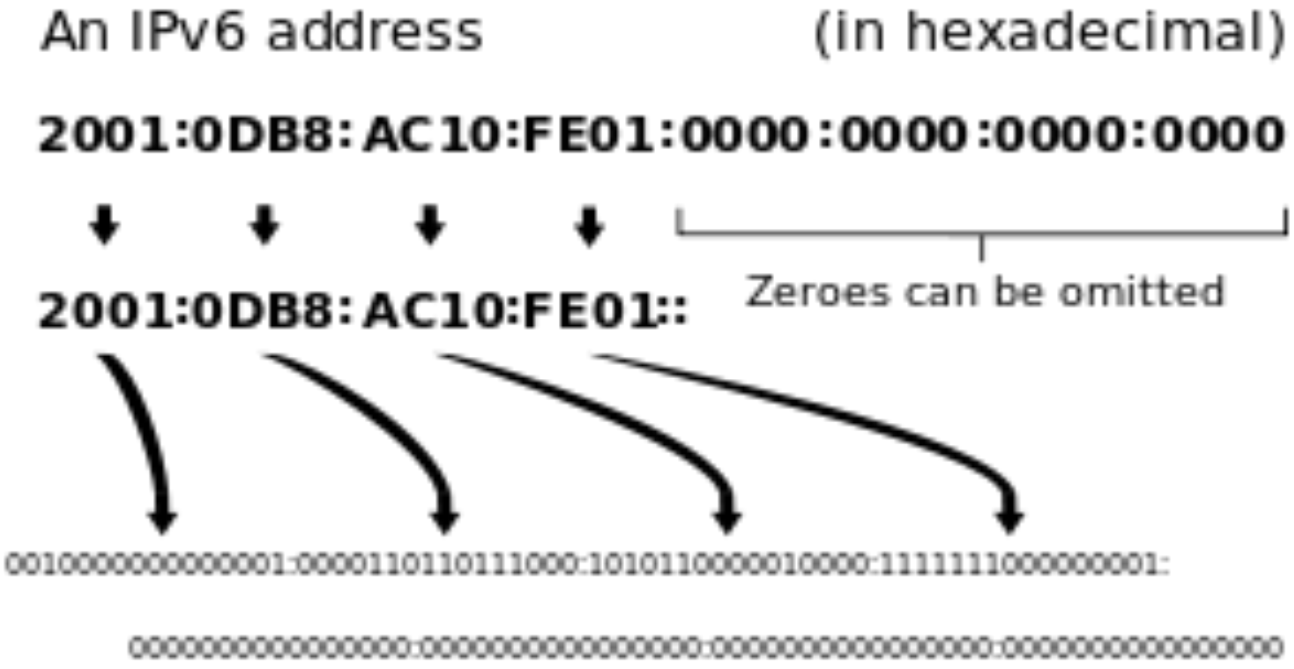
IPv6

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 3.4x10^{38} addresses.
- That’s 4.8x10^{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

Zeroes can be omitted

2001:0DB8:AC10:FE01::

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”

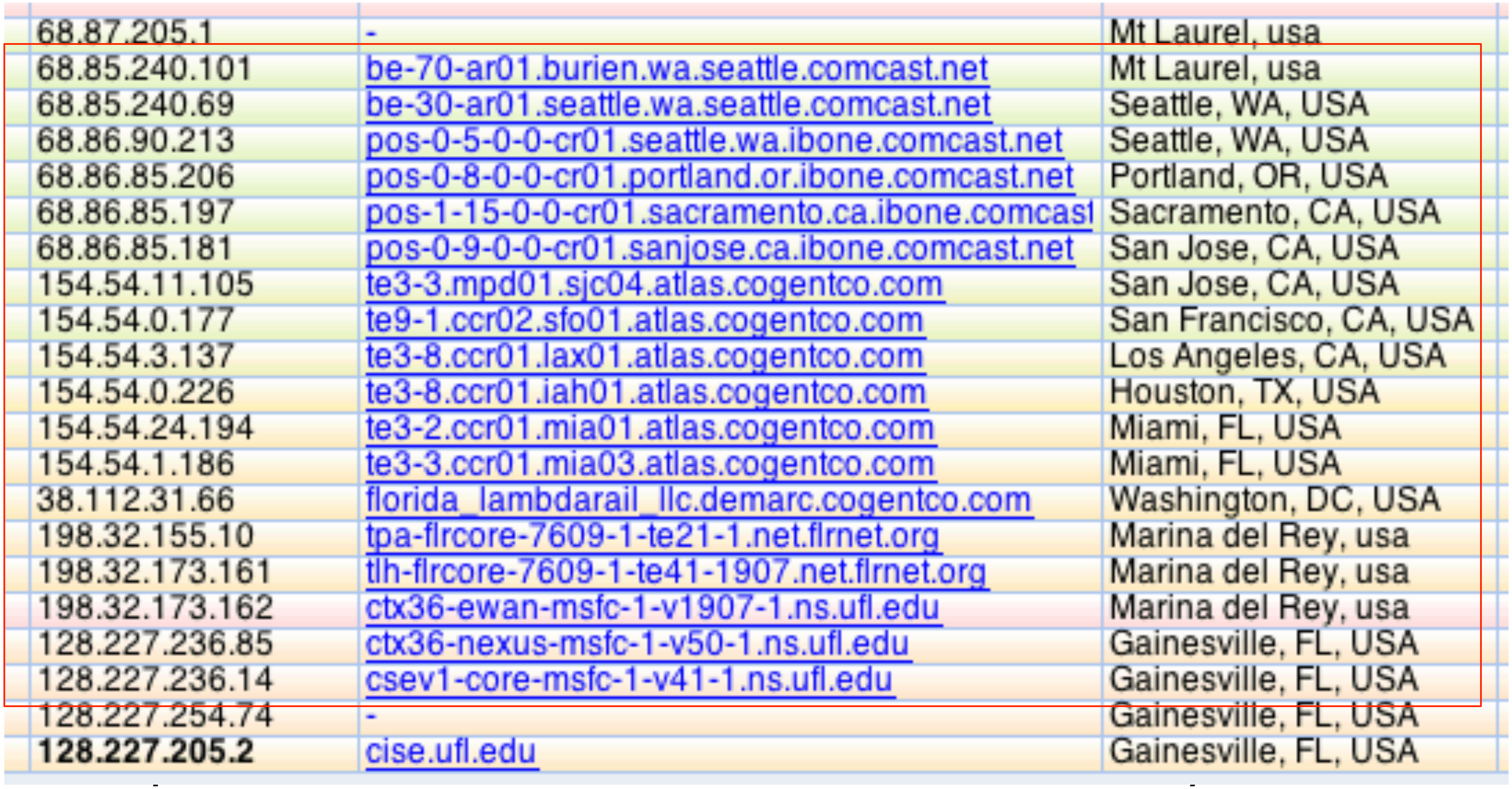
Domain Name System

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

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	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_lambdarail_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

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

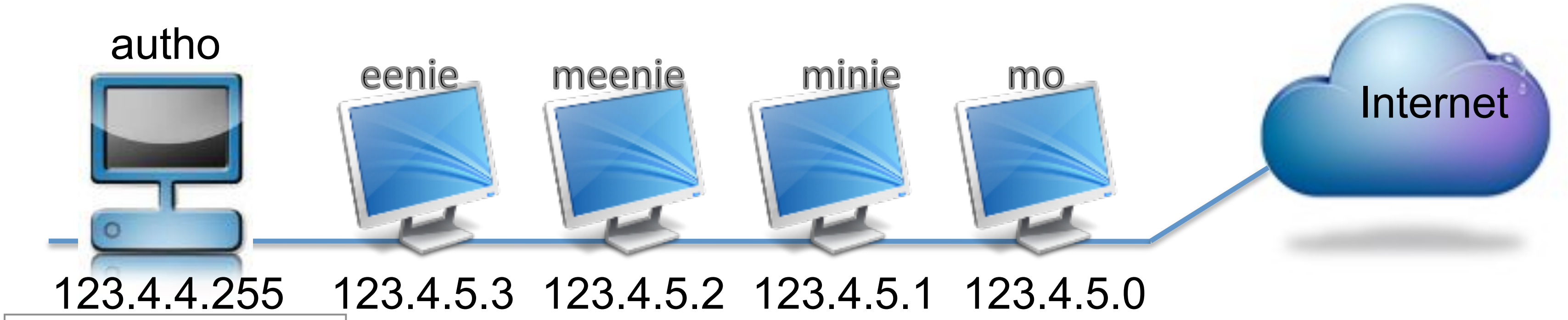
If It Has Never Been Asked …

- 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

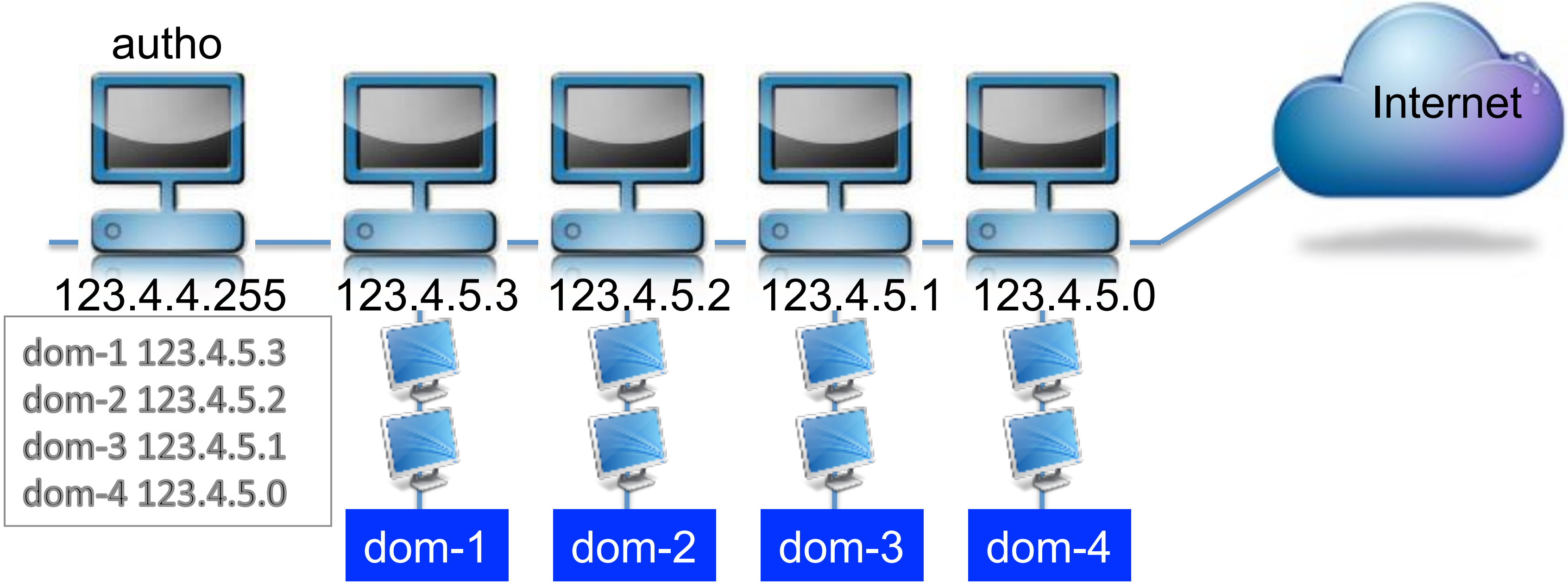


eenie	123.4.5.3
meenie	123.4.5.2
minie	123.4.5.1
mo	123.4.5.0

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

The DNS Design: Fact 1 (Continued)

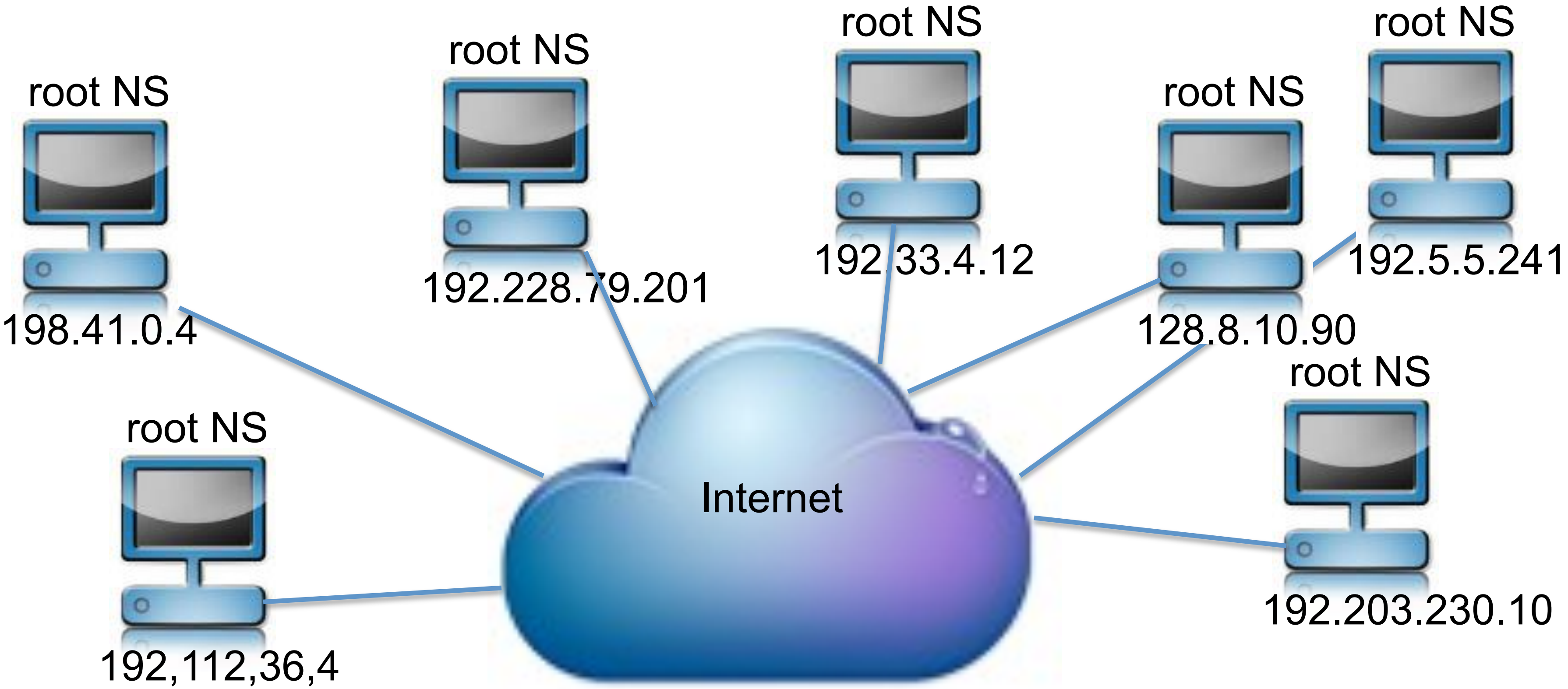
- OR Autho knows the number of every autho computer in its domain



dom-1	123.4.5.3
dom-2	123.4.5.2
dom-3	123.4.5.1
dom-4	123.4.5.0

The DNS Design: Fact 2

- There are 13 Internet "root name servers" scattered around the world … all the same



- All DNS servers have their numbers

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 …

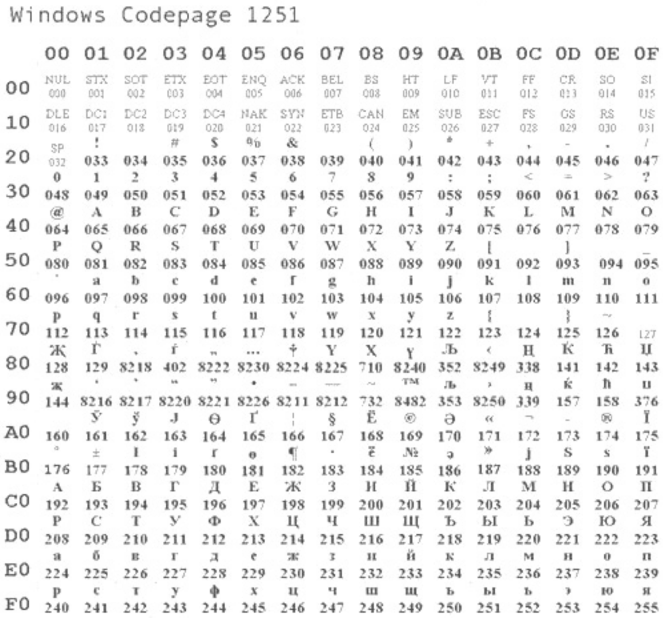
- Fault tolerant: when a hurricane takes out Miami's power, only the domains without power are affected …
- Robust: when a fire burns down the building of a .root name server, 12 others can carry the load
- Enormous capacity: most lookups are independent and do not collide (b/c higher level domain authos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

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

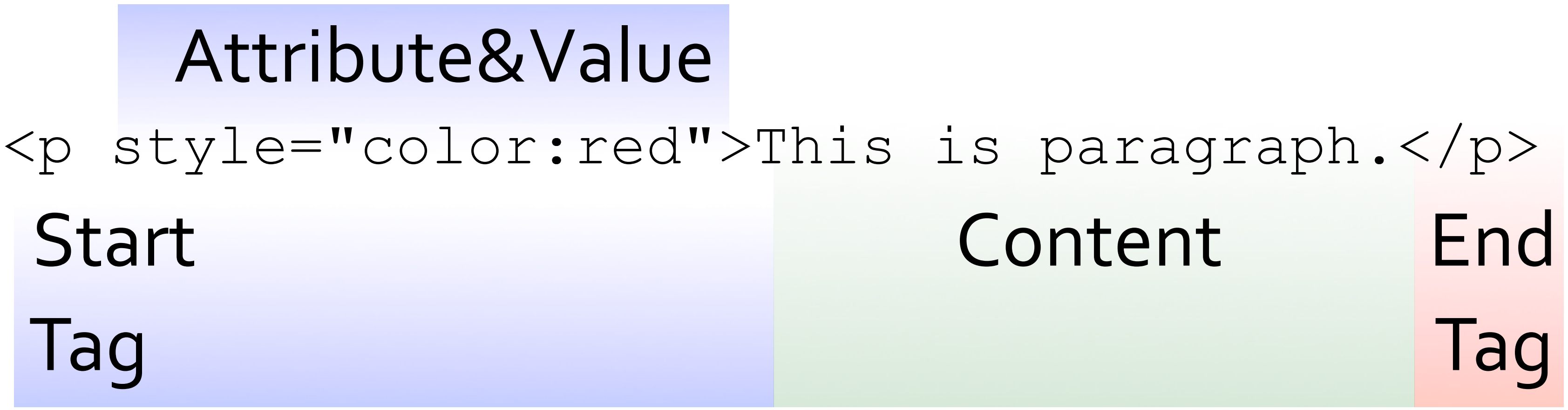
	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00	NUL 000	STX 001	SOT 002	ETX 003	EOT 004	ENQ 005	ACK 006	BEL 007	BS 008	HT 009	LF 010	VT 011	FF 012	CR 013	SO 014	SI 015
10	DLE 016	DC1 017	DC2 018	DC3 019	DC4 020	NAK 021	SYN 022	ETB 023	CAN 024	EM 025	SUB 026	ESC 027	FS 028	GS 029	RS 030	US 031
20	SP 032	! 033	034	# 035	$ 036	% 037	& 038	039	(040	) 041	* 042	+ 043	, 044	- 045	. 046	/ 047
30	0 048	1 049	2 050	3 051	4 052	5 053	6 054	7 055	8 056	9 057	: 058	; 059	< 060	= 061	> 062	? 063
40	@ 064	A 065	B 066	C 067	D 068	E 069	F 070	G 071	H 072	I 073	J 074	K 075	L 076	M 077	N 078	O 079
50	P 080	Q 081	R 082	S 083	T 084	U 085	V 086	W 087	X 088	Y 089	Z 090	[091	092	] 093	094	_ 095
60	` 096	a 097	b 098	c 099	d 100	e 101	f 102	g 103	h 104	i 105	j 106	k 107	l 108	m 109	n 110	o 111
70	p 112	q 113	r 114	s 115	t 116	u 117	v 118	w 119	x 120	y 121	z 122	{ 123	124	} 125	~ 126	127
80	Ж 128	Ѓ 129	‚ 8218	ѓ 402	„ 8222	… 8230	† 8224	Ү 8225	Х 710	ү 8240	Љ 352	‹ 8249	Ң 338	Ќ 141	Ћ 142	Џ 143
90	ж 144	' 8216	' 8217	" 8220	" 8221	• 8226	– 8211	— 8212	~ 732	™ 8482	љ 353	› 8250	ң 339	ќ 157	ћ 158	џ 376
A0	160	Ў 161	ў 162	Ј 163	Ө 164	Ґ 165	¦ 166	§ 167	Ё 168	© 169	Ә 170	« 171	¬ 172	- 173	® 174	Ї 175
B0	° 176	± 177	І 178	і 179	ґ 180	ө 181	¶ 182	· 183	ё 184	№ 185	ә 186	» 187	ј 188	Ѕ 189	ѕ 190	ї 191
C0	А 192	Б 193	В 194	Г 195	Д 196	Е 197	Ж 198	З 199	И 200	Й 201	К 202	Л 203	М 204	Н 205	О 206	П 207
D0	Р 208	С 209	Т 210	У 211	Ф 212	Х 213	Ц 214	Ч 215	Ш 216	Щ 217	Ъ 218	Ы 219	Ь 220	Э 221	Ю 222	Я 223
E0	а 224	б 225	в 226	г 227	д 228	е 229	ж 230	з 231	и 232	й 233	к 234	л 235	м 236	н 237	о 238	п 239
F0	р 240	с 241	т 242	у 243	ф 244	х 245	ц 246	ч 247	ш 248	щ 249	ъ 250	ы 251	ь 252	э 253	ю 254	я 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 Tag

Content

End 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

```
<img src=“cooking-ewan-isabel.jpg" alt=”Kids Cooking"/>
```

Tag Image Source Alt Description End

- 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.

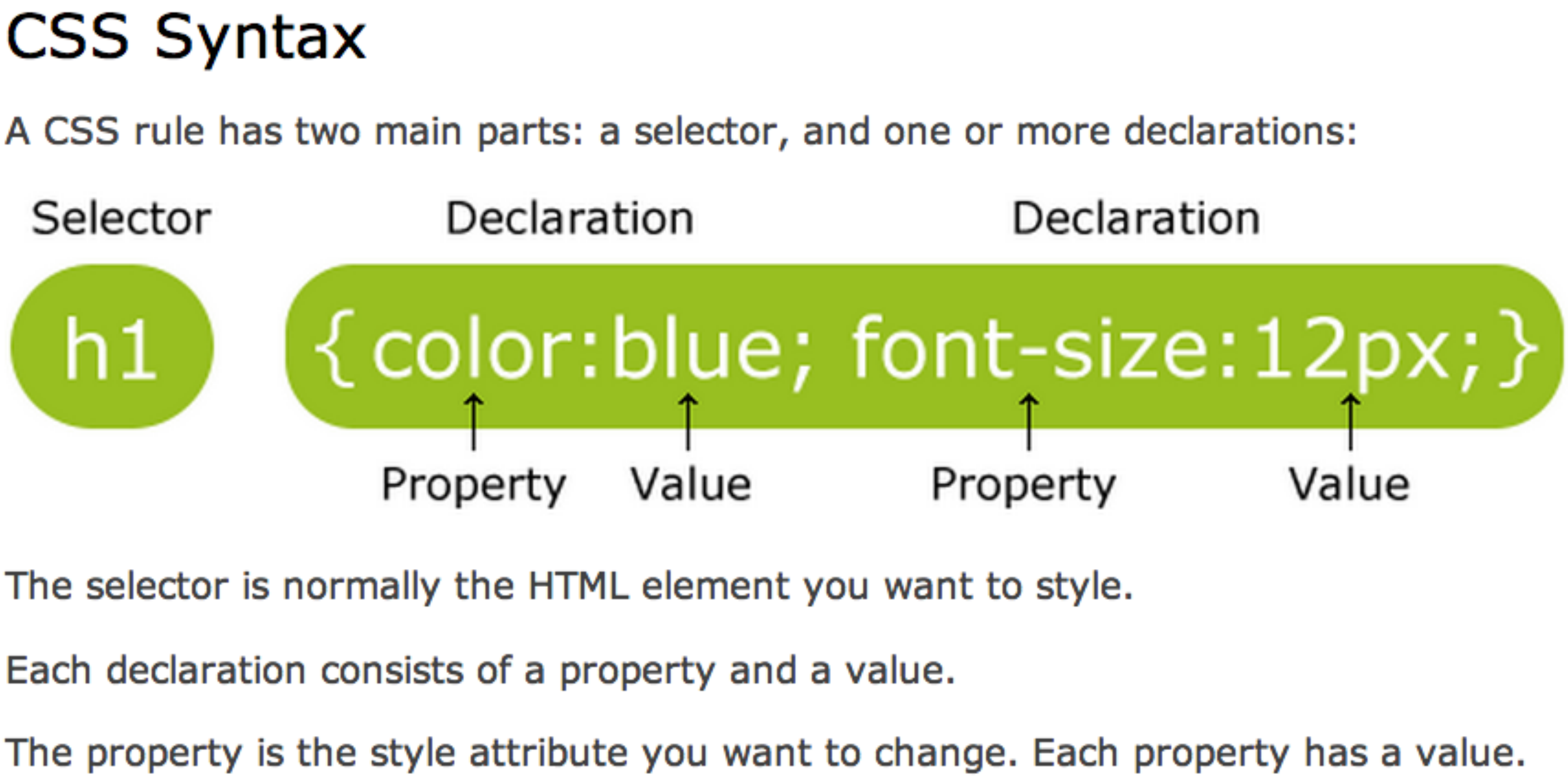
C. It is a language that allows for "documents" to be created that are not linear. (A book with chapters is linear – you normally read from start to end in order.)

D. All of A-C apply to HTML.

CSS – separating style from content

Content vs Style

- <h1>This is a heading</h1>
- <em>emphasize this</em>
- <b>Make this bold face</b>
- <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:

```
h1 {color:blue; font-size:12px;}
```

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="Goodbye!";
}
```

```
<!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

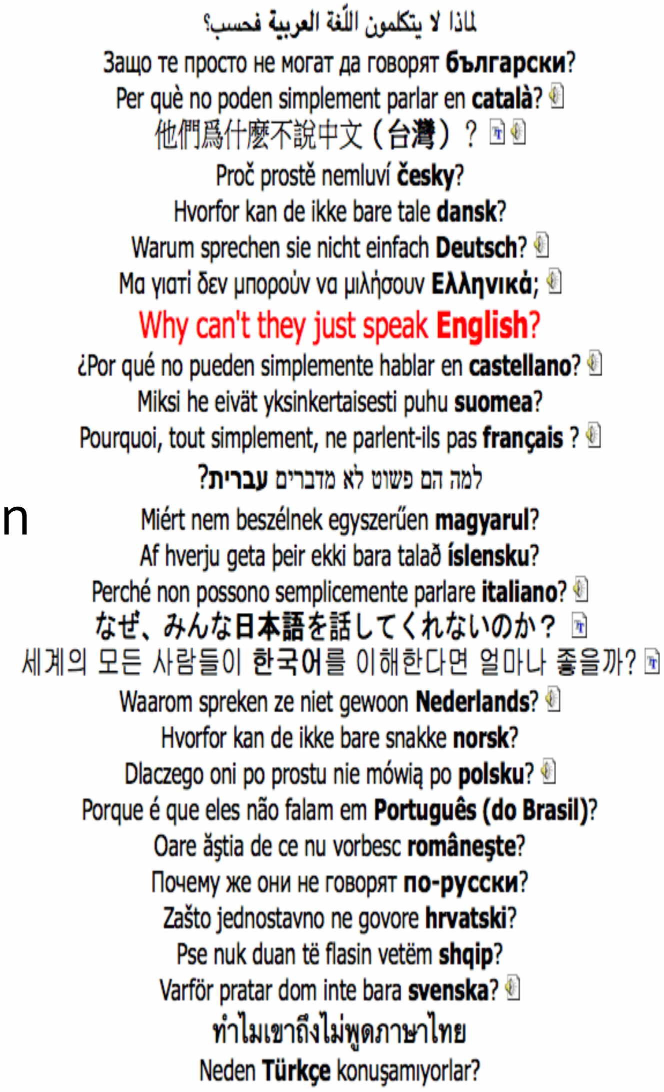
C
A
T

ASCII	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
0000	NU	SH	SX	EX	ET	EQ	AK	BL	BS	HT	LF	YT	FF	CR	SO	SI
0001	DL	D1	D2	D3	D4	NK	SY	EΣ	CN	EM	SB	EC	FS	GS	RS	US
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	{	|	}	~	DT
1000	80	81	82	83	IN	NL	SS	ES	HS	HJ	YS	PD	PV	RI	S2	S3
1001	DC	P1	PZ	SE	CC	MM	SP	EP	Q8	QQ	QA	CS	ST	OS	PM	AP
1010	AO	¡	¢	£	♀	¥	¦	§	¨	©	♂	«	¬	-	®	¯
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	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

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 اللّغة العربية فحسب؟
Защо 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 **한국어**를 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 ăştia de ce nu vorbesc **româneşte**?
Почему же они не говорят **по-русски**?
Zašto jednostavno ne govore **hrvatski**?
Pse nuk duan të flasin vetëm **shqip**?
Varför pratar dom inte bara **svenska**?
ทำไมเขาถึงไม่พูดภาษาไทย
Neden **Türkçe** konuşamıyorlar?

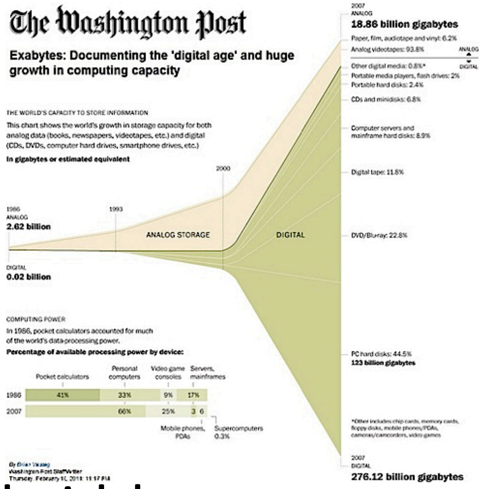
How many bits for all of Unicode?

There are 1,112,064 different Unicode characters. If a fixed bit format (like ascii with its 7 bits) where 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


The Washington Post

Exabytes: Documenting the 'digital age' and huge growth in computing capacity

THE WORLD'S CAPACITY TO STORE INFORMATION

This chart shows the world's growth in storage capacity for both analog data (books, newspapers, videotapes, etc.) and digital (CDs, DVDs, computer hard drives, smartphone drives, etc.)

In gigabytes or estimated equivalent

COMPUTING POWER

In 1986, pocket calculators accounted for much of the world's data-processing power.

Percentage of available processing power by device:

*Other includes chip cards, memory cards, floppy disks, mobile phones/PDAs, cameras/camcorders, video games

By Brian Vastag
Washington Post Staff Writer
Thursday, February 10, 2011; 11:17 PM

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 ⇒ 1/8×2 + 1/4×2 + 1/2×2 + 1/8×2 = 2 bits / symbol
  - Huffman ⇒ 1/8×3 + 1/4×2 + 1/2×1 + 1/8×3 = 1.75 bits / symbol

Huffman Code Example

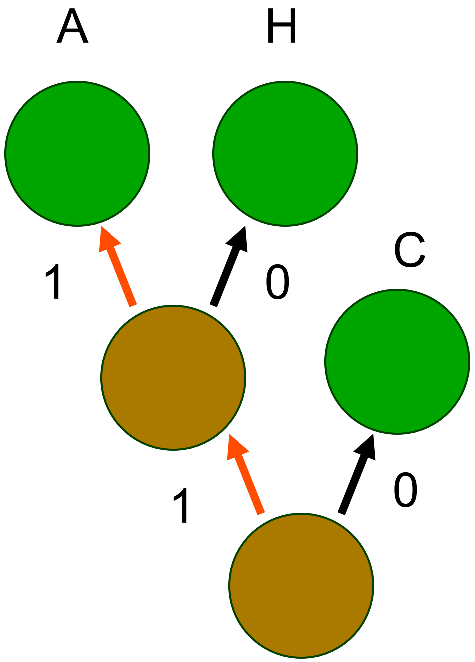
Symbol	Dog	Cat	Bird	Fish
Frequency	1/8	1/4	1/2	1/8
Original Encoding	00	01	10	11
	2 bits	2 bits	2 bits	2 bits
Huffman Encoding	110	10	0	111
	3 bits	2 bits	1 bit	3 bits

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

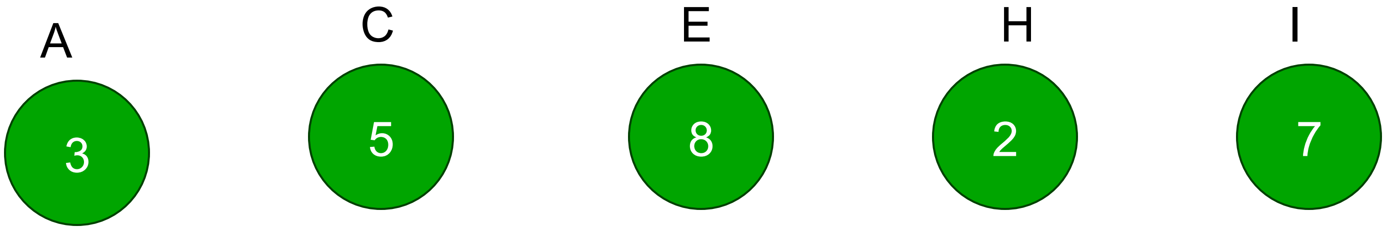
A. 0 B. 1 C. 2 D. 3 E. 4

Huffman Code Algorithm: Data Structures

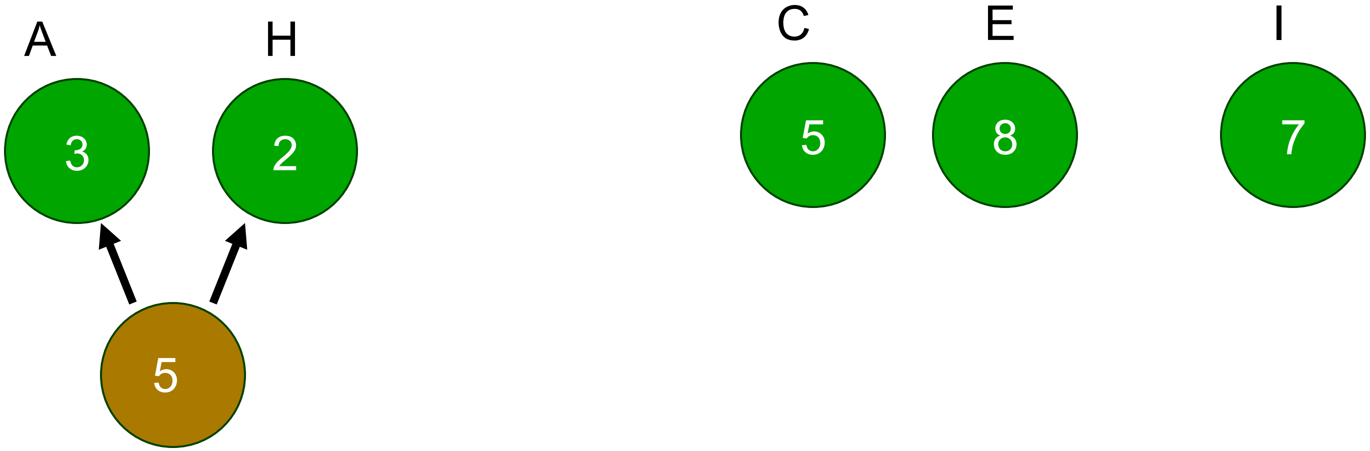
- Binary (Huffman) tree
  - Represents Huffman code
  - Edge ⇒ code (0 or 1)
  - Leaf ⇒ symbol
  - Path to leaf ⇒ encoding
  - Example
    - A = “11”, H = “10”, C = “0”
    - Good when ???
      - A, H less frequent than C in messages
- Want to efficiently build a binary tree



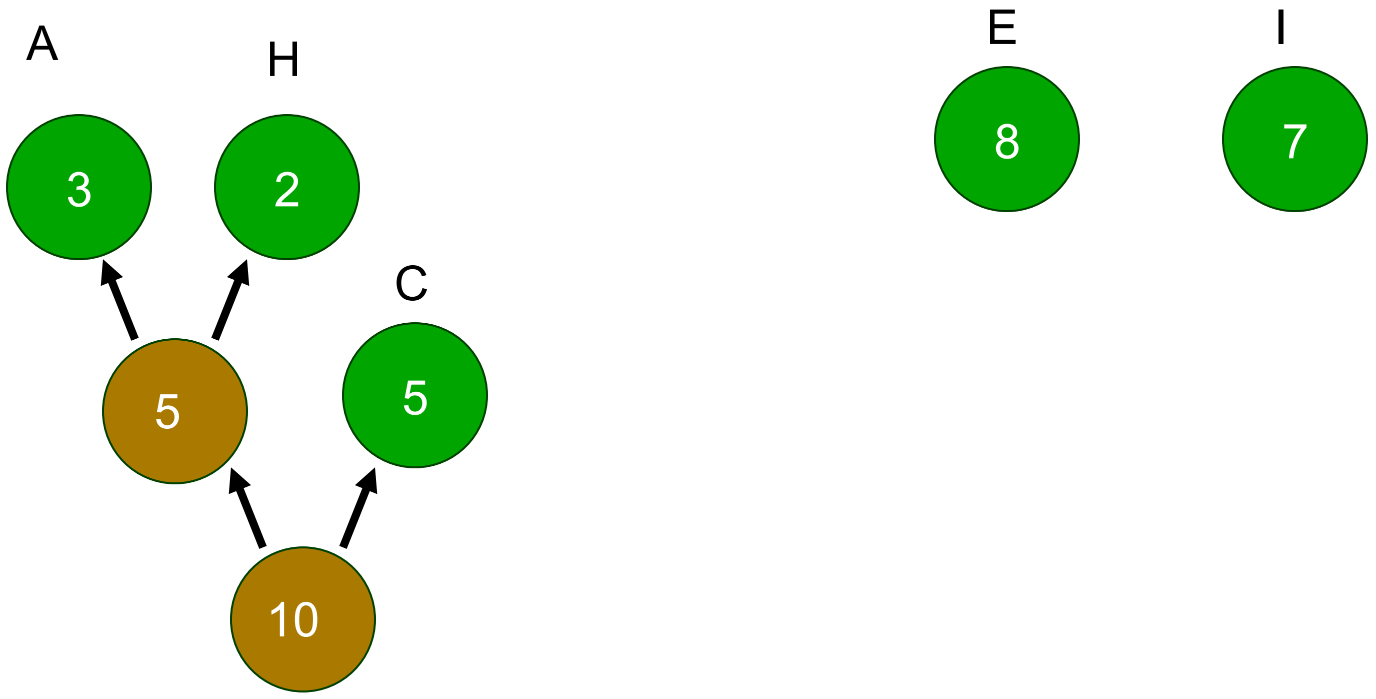
Huffman Tree Construction 1



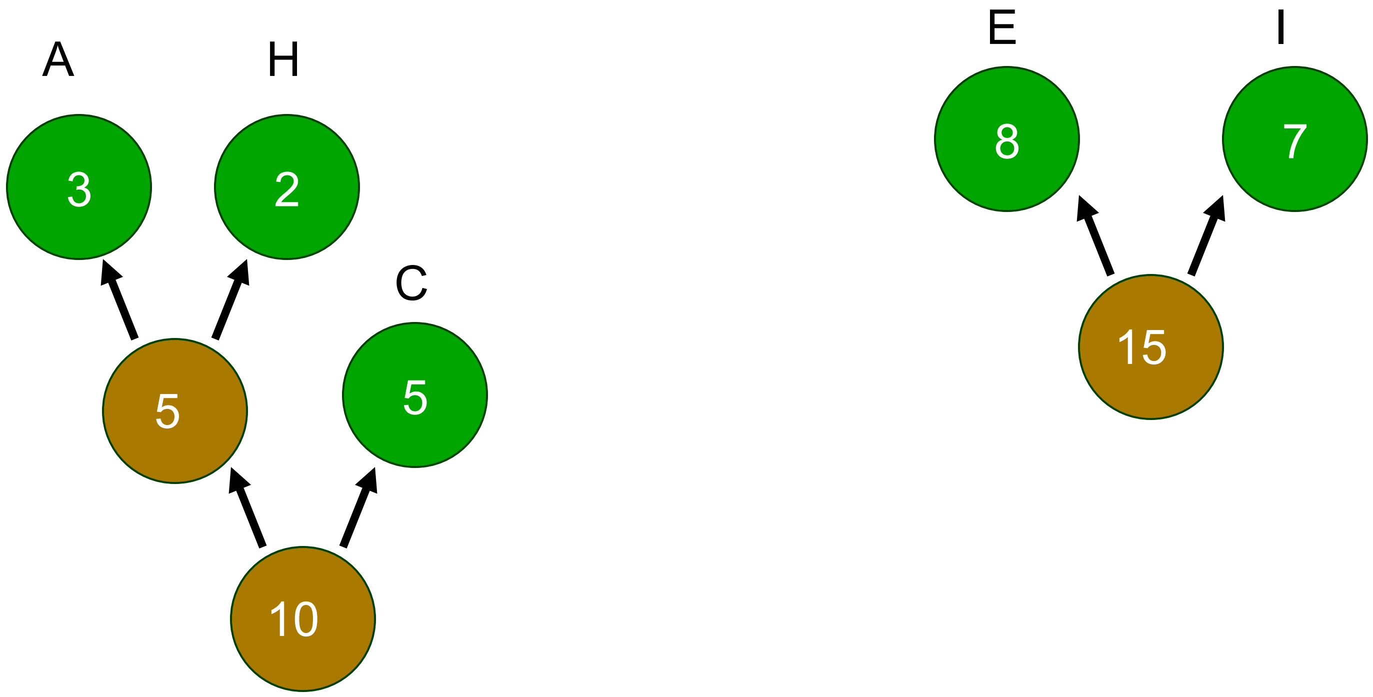
Huffman Tree Step 2: can first re-order by frequency



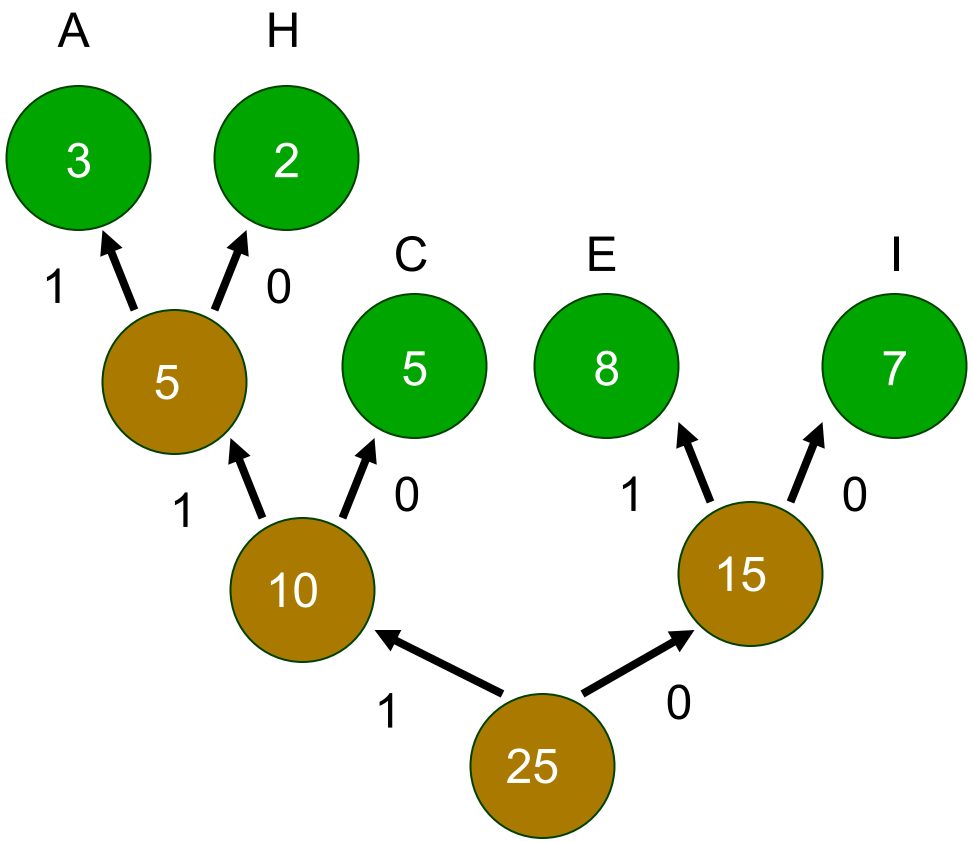
Huffman Tree Construction 3



Huffman Tree Construction 4



Huffman Tree Construction 5



E	= 01
I	= 00
C	= 10
A	= 111
H	= 110

Huffman Coding Example

- Huffman code

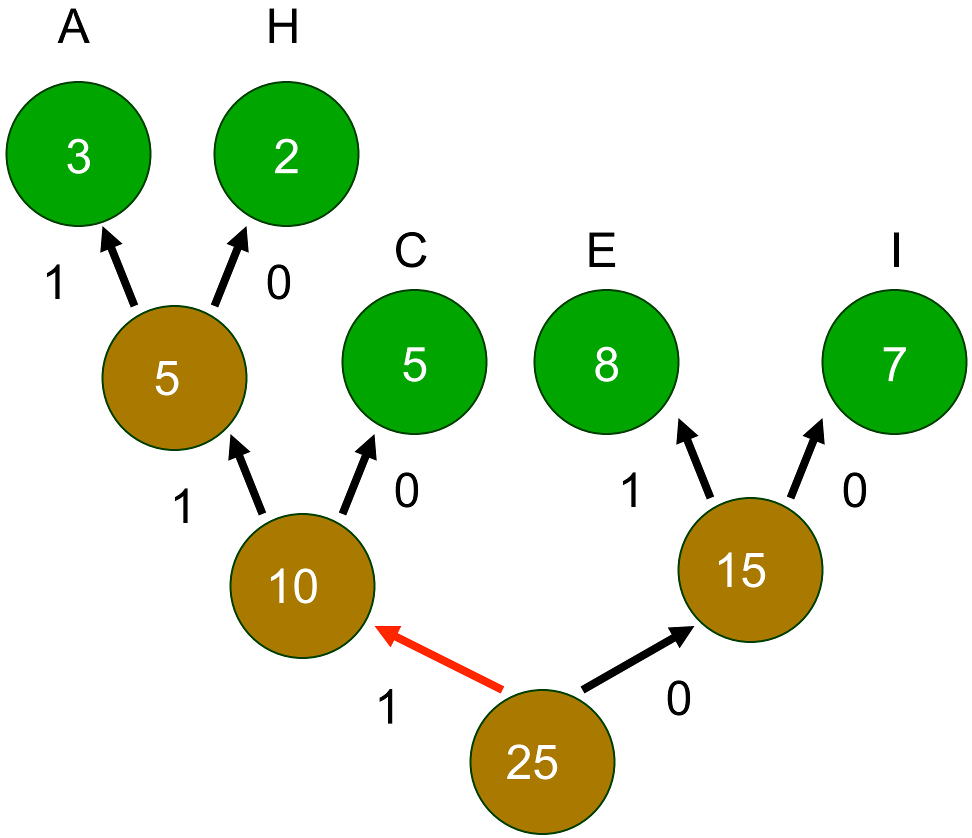
E	=	01
I	=	00
C	=	10
A	=	111
H	=	110

- Input
  - ACE
- Output
  - (111)(10)(01) = 111001

Huffman Code Algorithm Overview

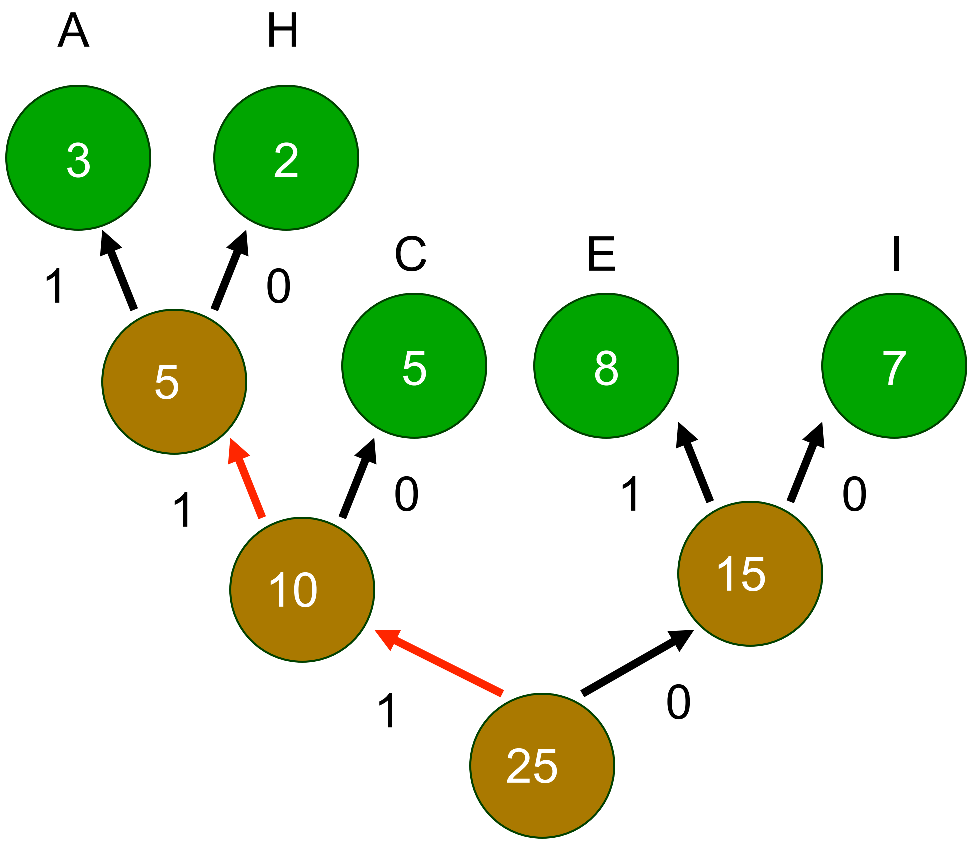
- Decoding
  - – Read compressed file & binary tree
  - – Use binary tree to decode file
    - Follow path from root to leaf

Huffman Decoding I



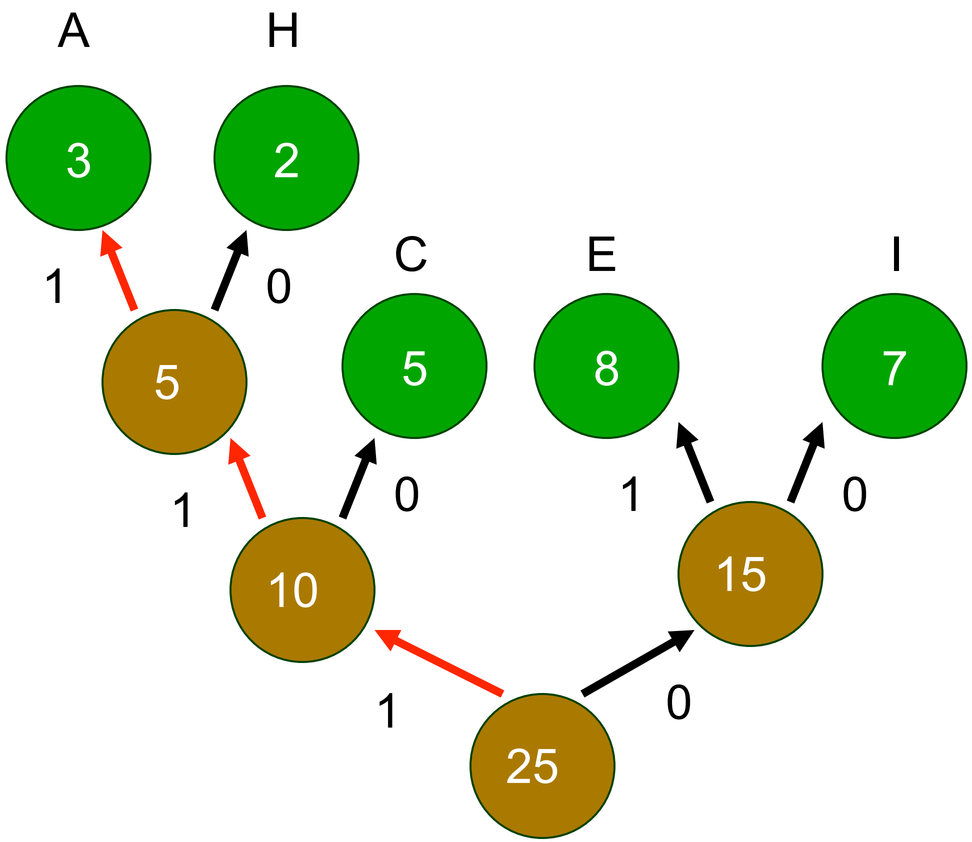
1111001

Huffman Decoding 2



11111001

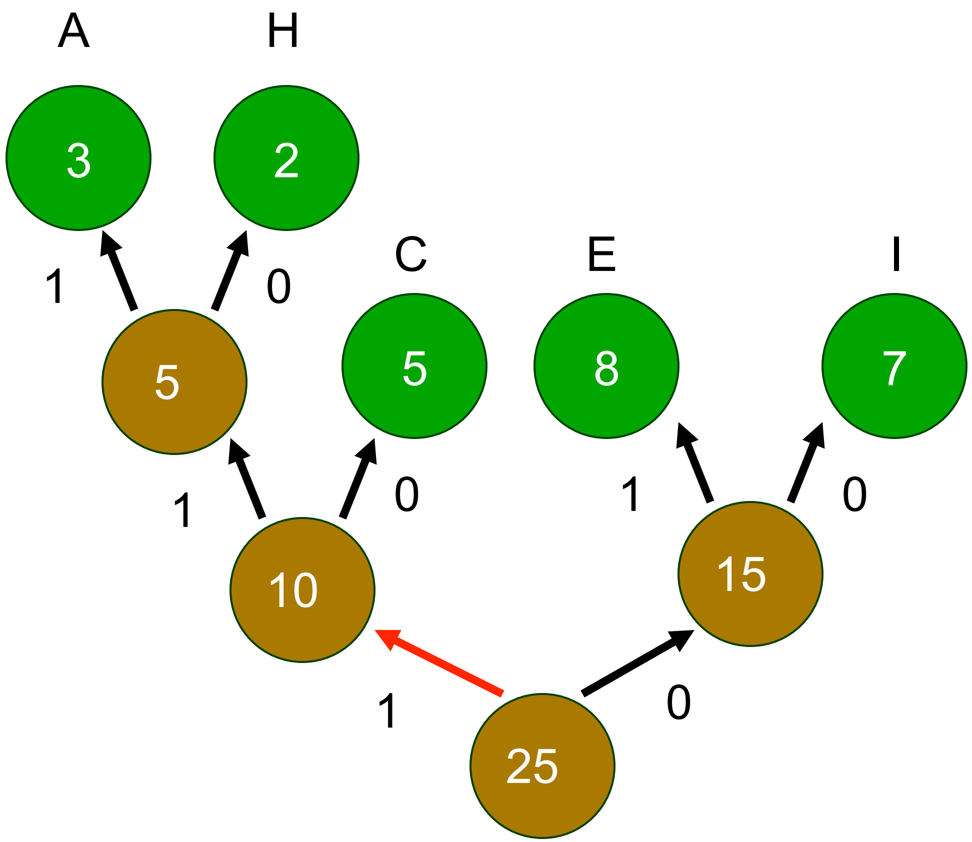
Huffman Decoding 3



1111001

A

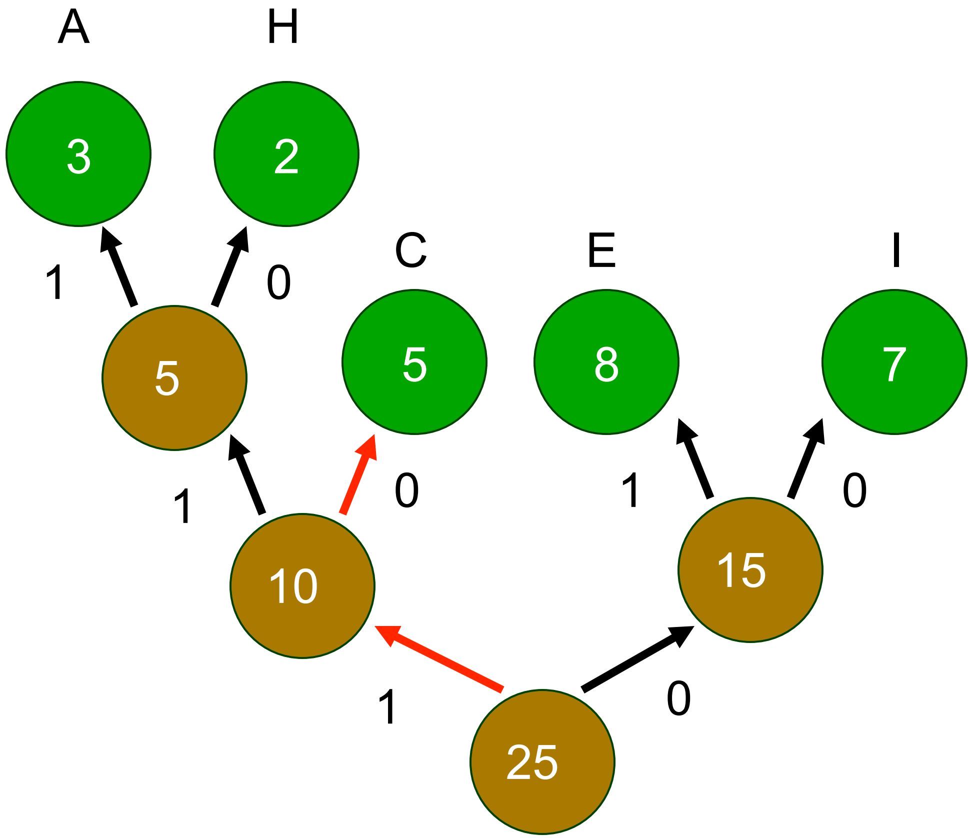
Huffman Decoding 4



1111001

A

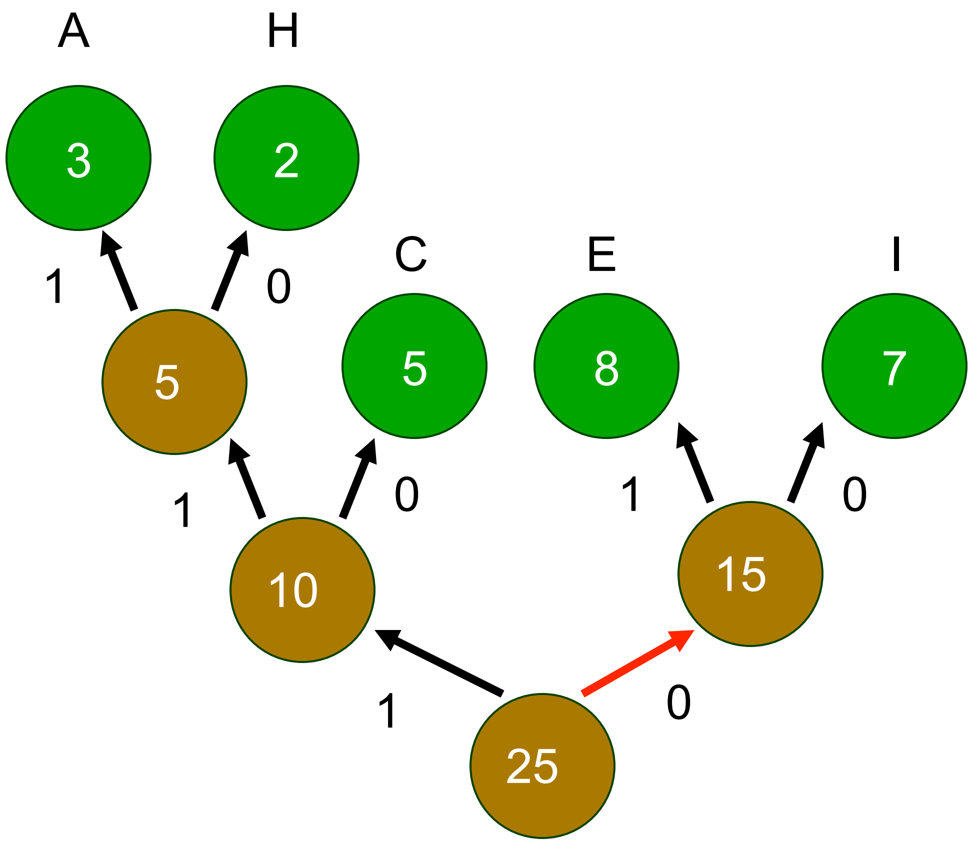
Huffman Decoding 5



1111001

AC

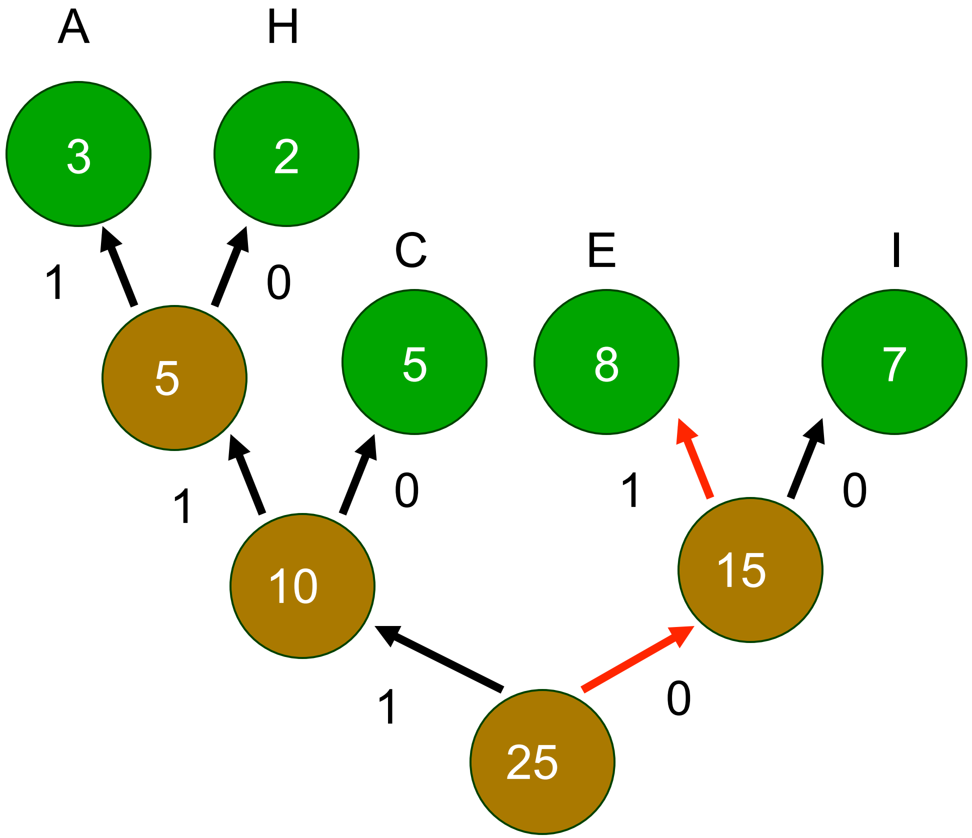
Huffman Decoding 6



1111001

AC

Huffman Decoding 7



1111001

ACE

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 (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*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 their 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. Unsolvable (like the halting problem)

Good Luck on the Final!