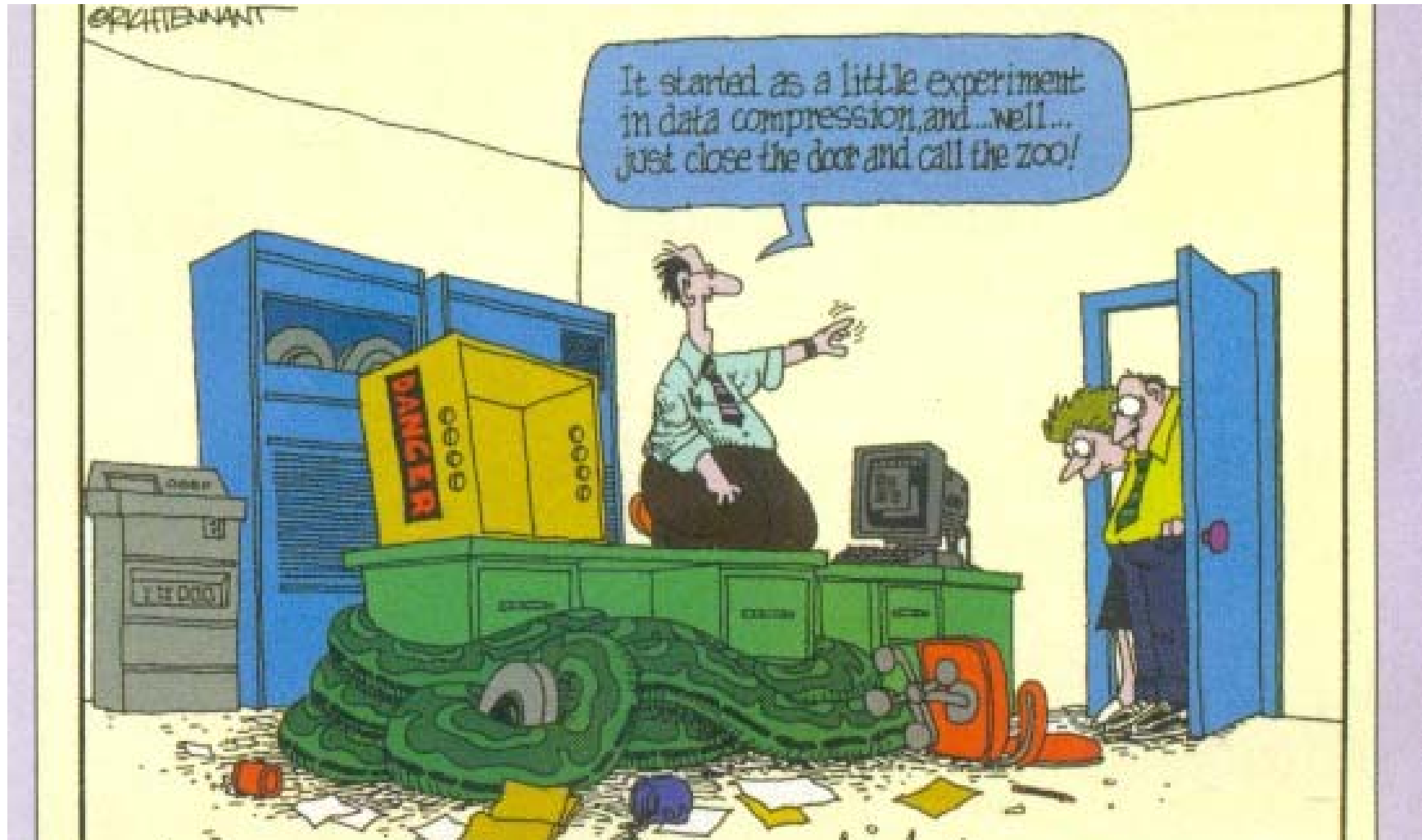


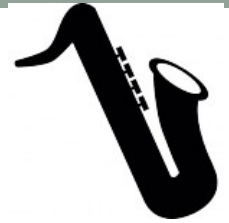


CSE 100: HUFFMAN CODES

Doing data compression the right way...



The Data Compression Problem



Digitize

Store

Communicate

11001110	01110000	01100111	00000011	11110000	11111111	01010001	11100011
00000101	11101111	00001000	01001011	11011110	00010000	10010111	10111100
00100001	00101111	01111000	01000010	01011110	11110000	01000110	01111100
00001111	10110111	00000011	01111110	10100001	00011100	01100100	11001100
00011011	11001111	11111110	11100011	11000001	00001011	11011011	01011101
11001010	11101111	10010010	10010101	11011111	00100100	00101001	00011010
01010011	00110000	01010100	01010011	11101100	01100111	10111010	01011001
01000011	11101111	00010111	11010111	01001011	10111010	00011111	11110000
10001100	10010110	00001010	00010000	10011000	00111011	00100000	11011100
00011010	01101101	00000011	11001100	11001100	11001110	01111001	01101101
10010100	10001101	01100001	01011010	01010010	10111011	10010101	11011100

But alas this lovely text must
be decomposed to bits.

Good news is: if we
decompose well, we can
recover it 😊

RIGHT HO, JEEVES

By
P. G. WODEHOUSE

How to represent in bits?

1922

To
RAYMOND NEEDHAM, K.C.
WITH AFFECTION AND ADMIRATION

-1-

"Jeeves," I said, "may I speak frankly?"

"Certainly, sir."

"What I have to say may wound you."

"Not at all, sir."

"Well, then——"

No—wait. Hold the line a minute. I've gone off the rails.

I don't know if you have had the same experience, but the snag I always come up against when I'm telling a story is this dashed difficult problem of where to begin it. It's a thing you don't want to go wrong over, because one false step and you're sunk. I mean, if you fool about too long at the start, trying to establish atmosphere, as they call it, and all that sort of rot, you fail to grip and the customers walk out on you.

Get off the mark, on the other hand, like a scalded cat, and your public is at a loss. It simply raises its eyebrows, and can't make out what you're talking about.

And in opening my report of the complex case of Gussie Fink-Nottle, Madeline Bassett, my Cousin Angela, my Aunt Dahlia, my Uncle Thomas, young Tuppy Glossop and the cook, Anatole, with the above spot of dialogue, I see that I have made the second of these two floaters.

I shall have to hark back a bit. And taking it for all in all and weighing this against that, I suppose the affair may be said to have had its inception, if inception is the word I want, with that visit of mine to Cannes. If I hadn't gone to Cannes, I shouldn't have met the Bassett or bought that white mess jacket, and Angela wouldn't have met her shark, and Aunt Dahlia wouldn't have played baccarat.

Yes, most decidedly, Cannes was the *point d'appui*.

Right ho, then. Let me marshal my facts.

I went to Cannes—leaving Jeeves behind, he having intimated that he did not wish to miss Ascot—round about the beginning of June. With me travelled my Aunt Dahlia and her daughter Angela. Tuppy Glossop, Angela's betrothed, was to have been of the party, but at the last moment couldn't get away. Uncle Tom, Aunt Dahlia's husband, remained at home, because he can't stick the South of France at any price.

How to represent in bits?

- Encoding with ASCII
- Symbols and dictionaries

How many bits do we need?

RIGHT HO, JEEVES

By

P. G. WODEHOUSE

RAYMOND
WITH AFFECTIO

"Jeeves," I said, "may I speak frankly?"

"Certainly, sir."

"What I have to say may wound you."

"Not at all, sir."

"Well, then——"

No—wait. Hold the line a minute. I've gone off the rails.

I don't know if you have had the same experience, but the snag I always to begin it. It's a thing you don't want to go wrong over, because one fal establish atmosphere, as they call it, and all that sort of rot, you fail to grip

Get off the mark, on the other hand, like a scalded cat, and your public about.

And in opening my report of the complex case of Gussie Fink-Nottle, I Tuppy Glossop and the cook, Anatole, with the above spot of dialogue, I se

I shall have to hark back a bit. And taking it for all in all and weighin inception is the word I want, with that visit of mine to Cannes. If I hadn't , and Angela wouldn't have met her shark, and Aunt Dahlia wouldn't have p

Yes, most decidedly, Cannes was the *point d'appui*.

Right ho, then. Let me marshal my facts.

I went to Cannes—leaving Jeeves behind, he having intimated that he d my Aunt Dahlia and her daughter Angela. Tuppy Glossop, Angela's betroth Tom, Aunt Dahlia's husband, remained at home, because he can't stick the

The ASCII table

Dec	Hex	Oct	Chr	Dec	Hex	Oct	HTML	Chr	Dec	Hex	Oct	HTML	Chr	Dec	Hex	Oct	HTML	Chr
0	0	000	NULL	32	20	040	 Space	64	40	100	@ @	96	60	140	` `			
1	1	001	Start of Header	33	21	041	! !	65	41	101	A A	97	61	141	a a			
2	2	002	Start of Text	34	22	042	" "	66	42	102	B B	98	62	142	b b			
3	3	003	End of Text	35	23	043	# #	67	43	103	C C	99	63	143	c c			
4	4	004	End of Transmission	36	24	044	$ \$	68	44	104	D D	100	64	144	d d			
5	5	005	Enquiry	37	25	045	% %	69	45	105	E E	101	65	145	e e			
6	6	006	Acknowledgment	38	26	046	& &	70	46	106	F F	102	66	146	f f			
7	7	007	Bell	39	27	047	' '	71	47	107	G G	103	67	147	g g			
8	8	010	Backspace	40	28	050	((	72	48	110	H H	104	68	150	h h			
9	9	011	Horizontal Tab	41	29	051	))	73	49	111	I I	105	69	151	i i			
10	A	012	Line feed	42	2A	052	* *	74	4A	112	J J	106	6A	152	j j			
11	B	013	Vertical Tab	43	2B	053	+ +	75	4B	113	K K	107	6B	153	k k			
12	C	014	Form feed	44	2C	054	, ,	76	4C	114	L L	108	6C	154	l l			
13	D	015	Carriage return	45	2D	055	- -	77	4D	115	M M	109	6D	155	m m			
14	E	016	Shift Out	46	2E	056	. .	78	4E	116	N N	110	6E	156	n n			here
15	F	017	Shift In	47	2F	057	/ /	79	4F	117	O O	111	6F	157	o o			ng to
16	10	020	Data Link Escape	48	30	060	0 0	80	50	120	P P	112	70	160	p p			
17	11	021	Device Control 1	49	31	061	1 1	81	51	121	Q Q	113	71	161	q q			
18	12	022	Device Control 2	50	32	062	2 2	82	52	122	R R	114	72	162	r r			lking
19	13	023	Device Control 3	51	33	063	3 3	83	53	123	S S	115	73	163	s s			
20	14	024	Device Control 4	52	34	064	4 4	84	54	124	T T	116	74	164	t t			oung
21	15	025	Negative Ack.	53	35	065	5 5	85	55	125	U U	117	75	165	u u			
22	16	026	Synchronous idle	54	36	066	6 6	86	56	126	V V	118	76	166	v v			
23	17	027	End of Trans. Block	55	37	067	7 7	87	57	127	W W	119	77	167	w w			on, if
24	18	030	Cancel	56	38	070	8 8	88	58	130	X X	120	78	170	x x			cket,
25	19	031	End of Medium	57	39	071	9 9	89	59	131	Y Y	121	79	171	y y			
26	1A	032	Substitute	58	3A	072	: :	90	5A	132	Z Z	122	7A	172	z z			
27	1B	033	Escape	59	3B	073	; ;	91	5B	133	[[	123	7B	173	{ {			
28	1C	034	File Separator	60	3C	074	< <	92	5C	134	\ \	124	7C	174	|			
29	1D	035	Group Separator	61	3D	075	= =	93	5D	135	]]	125	7D	175	} }			elled
30	1E	036	Record Separator	62	3E	076	> >	94	5E	136	^ ^	126	7E	176	~ ~			ncle
31	1F	037	Unit Separator	63	3F	077	? ?	95	5F	137	_ _	127	7F	177	 Del			

Fixed length encoding

- Fixed length: each symbol is represented using a fixed number of bits
- For example if the symbols were 's', 'p', 'a', 'm' we might define the following encoding:

spamspamspam spamspamspam	
------------------------------	--

Text file

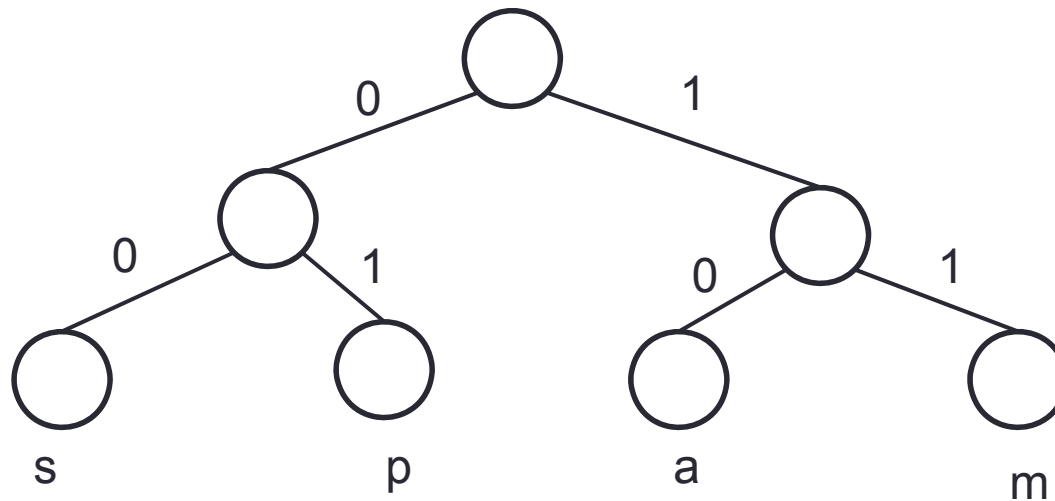
Encoded file

Symbol	Code word
s	00
p	01
a	10
m	11

For a dictionary consisting of M symbols, what is the minimum number of bits needed to encode each symbol (assume fixed length binary codes) ?

- A. 2^M B. M C. $M/2$ D. $\text{ceil}(\log_2 M)$ E. None of these

Binary codes as Binary Trees



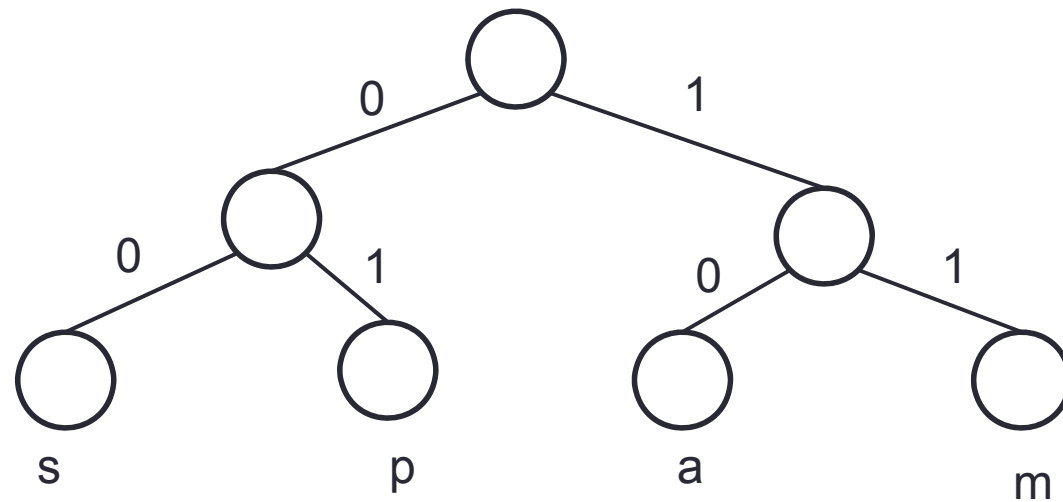
Code A

Symbol	Codeword
s	00
p	01
a	10
m	11

Symbols are leaf nodes

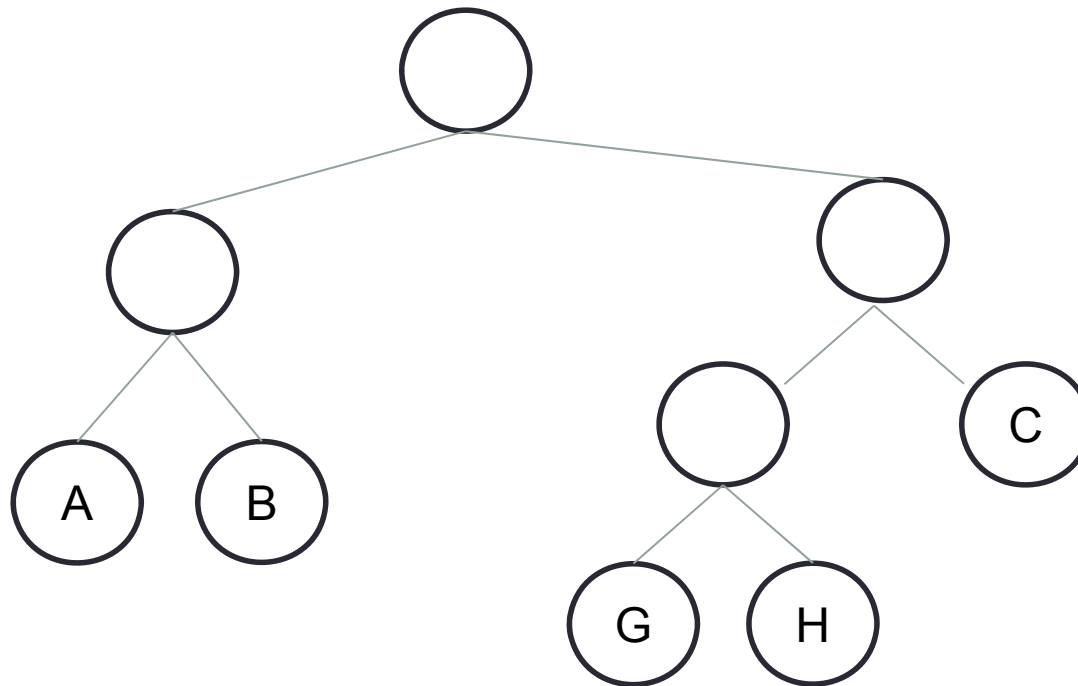
Root to leaf node gives the codeword for each symbol

Encoding with Binary Trees



Given the above binary tree, encode the string “papa”

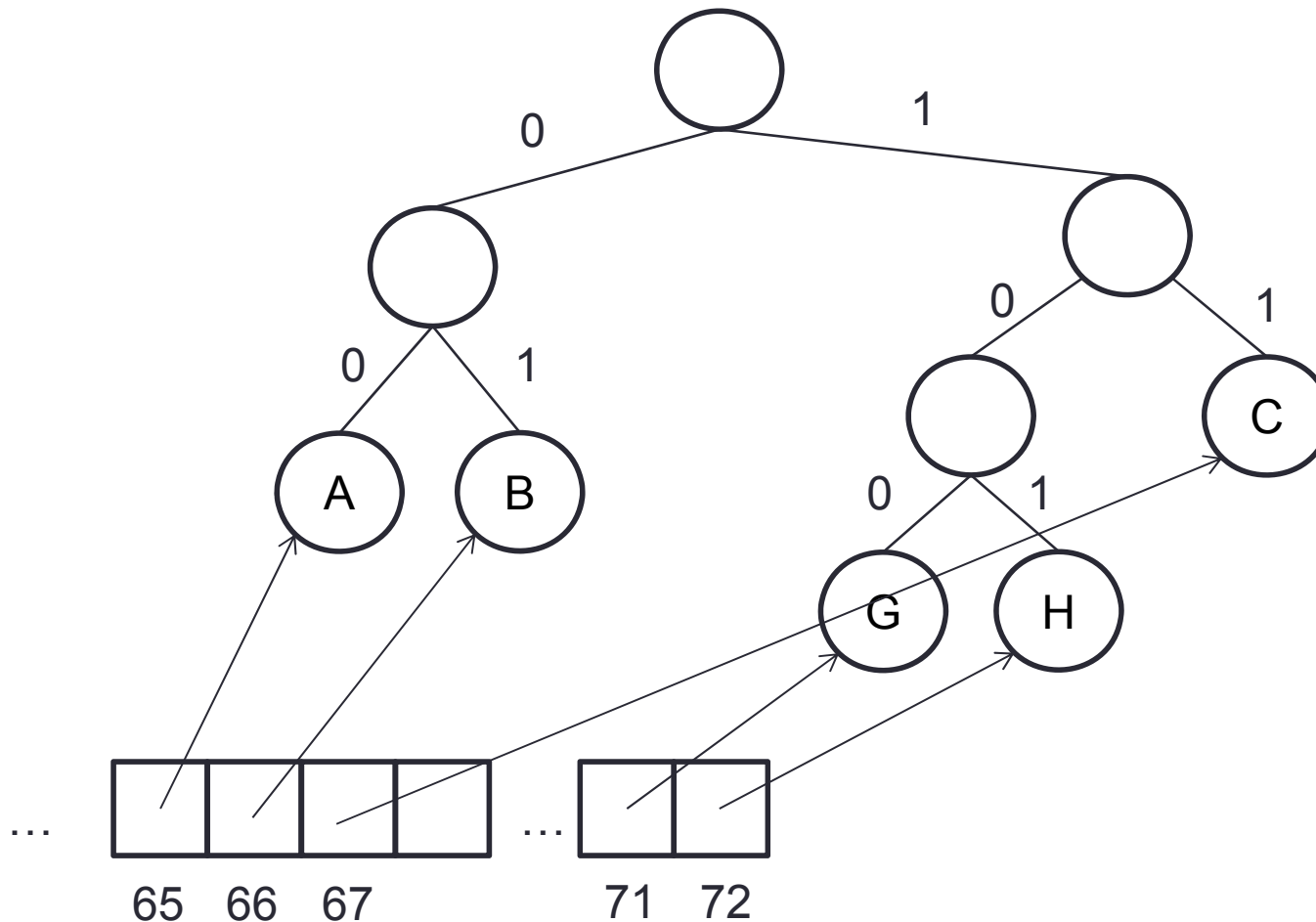
Encoding a symbol- think implementation!



A very bad way is to start at the root and search down the tree until you find the symbol you are trying to encode.

Encoding a symbol

A much better way is to maintain a list of leaves and then to traverse the tree to the root (and then reverse the code)



```
vector<HCNode*> leaves;
```

```
...
```

```
leaves = vector<HCNode*>(256, (HCNode*)0);
```

Traversing a list, with recursion

```
class LNode {  
    int data;  
    LNode* next;  
}
```



```
void traverse(LNode* n) {  
    // 1  
    if (n == nullptr) return;  
    // 2  
    traverse(n->next);  
    // 3  
}
```

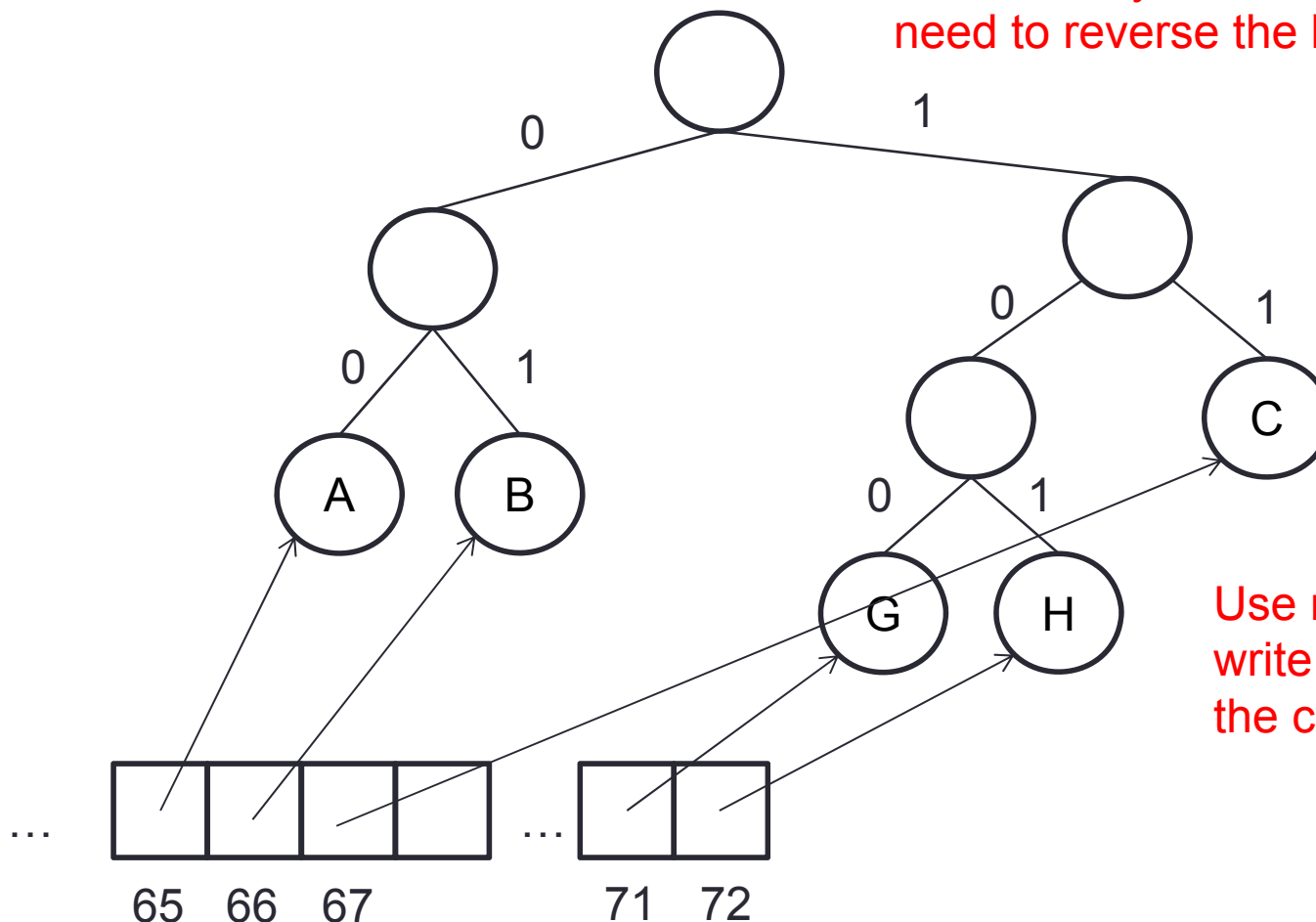
Where should I put the line to print `n->data` to print the list in *reverse order*?

```
std::cout << n->data << std::endl;
```

A. 1 B. 2 C. 3

Encoding a symbol

A much better way is to maintain a list of leaves and then to traverse the tree to the root. **If you use recursion, there's no need to reverse the list!**



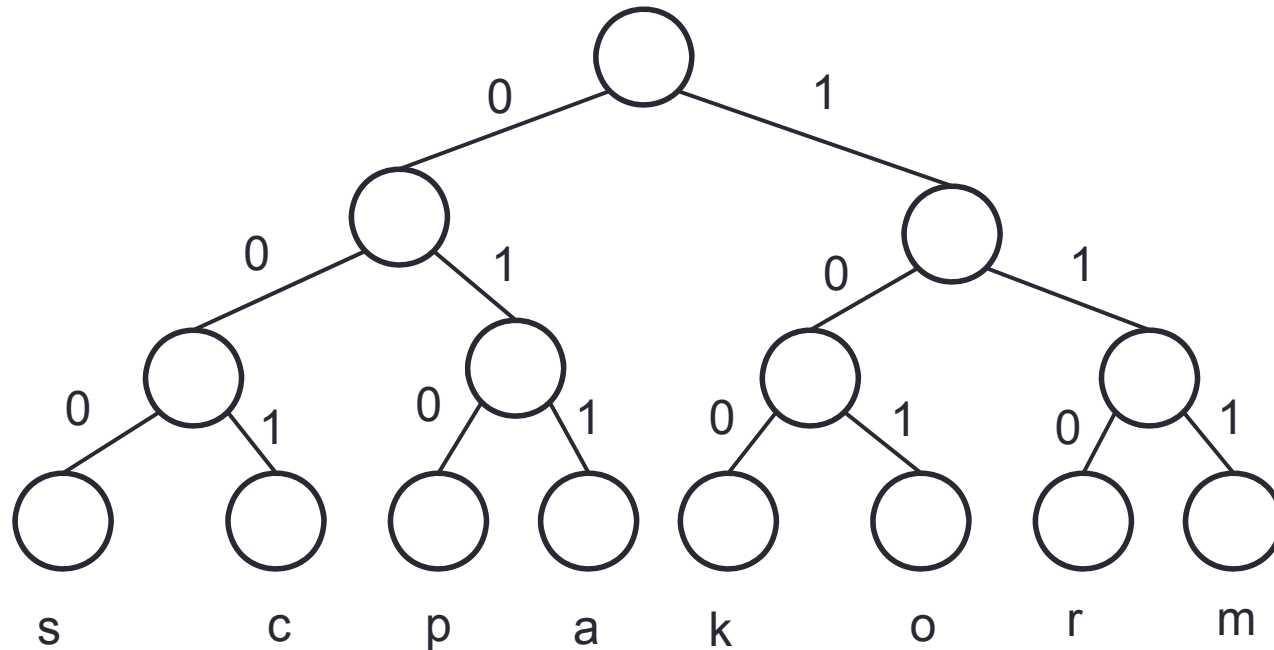
Use recursion to easily write a symbol's code in the correct order!

```
vector<HCNode*> leaves;
```

```
...
```

```
leaves = vector<HCNode*>(256, (HCNode*)0);
```

Decoding on binary trees



Decode the bitstream 110101001100 using the given binary tree

- A. scam
- B. mork
- C. rock
- D. korp

Is there an alternative to fixed length encoding?

```
=====
Composer:      Trad.
Author:        Eric Idle

Virgin Records 1989
MONT D1
=====

=====
  ALWAYS LOOK ON THE BRIGHT SIDE OF LIFE
=====

Some things in life are bad
They can really make you mad
Other things just make you swear and curse
When you're chewing on life's gristle
Don't grumble, give a whistle
And this'll help things turn out for the best ...
And ...
(music slides into the song)

... Always look on the bright side of life ...
(whistle)
... Always look on the bright side of life ...
(whistle)

If life seems jolly rotten
There's something you've forgotten
And that's to laugh and smile and dance and sing
When you're feeling in the dumps
Don't be silly chumps
Just purse your lips and whistle _ that's the thing.
And ...

... Always look on the bright side of life ...
(whistle)
Come on.
(others start to join in)
... Always look on the bright side of life ...
(whistle)

For life is quite absurd
And death's the final word
You must always face the curtain with a bow
Forget about your sin _
Give the audience a grin
Enjoy it _ it's your last chance anyhow.

So always look on the bright side of death
Just before you draw your terminal breath
```

- Can we go beyond fixed length encoding?
- What if certain symbols appeared more often than others?

Variable length codes

ssssssssssssssssss
sspppppaampamm

Text file

Symbol	Counts
--------	--------

s	18
p	6
a	3
m	3

Symbol	Frequency
--------	-----------

s	0.6
p	0.2
a	0.1
m	0.1

Code A

Symbol	Codeword
--------	----------

s	00
p	01
a	10
m	11

Code B

Symbol	Codeword
--------	----------

s	0
p	1
a	10
m	11

Is code B better than code A?

A. Yes

B. No

C. Depends

Comparing encoding schemes

ssssssssssssssssss
ssppppaampamm

Text file

Symbol	Frequency
s	0.6
p	0.2
a	0.1
m	0.1

Code A

Symbol	Codeword
s	00
p	01
a	10
m	11

Code B

Symbol	Codeword
s	0
p	1
a	10
m	11

Average length (code A) = 2 bits/symbol

Average length (code B) = $0.6 * 1 + 0.2 * 1 + 0.1 * 2 + 0.1 * 2$
= 1.2 bits/symbol

Decoding variable length codes

ssssssssssssssssss
sspppppaampamm

Text file

Symbol	Frequency
s	0.6
p	0.2
a	0.1
m	0.1

Code A

Symbol	Codeword
s	00
p	01
a	10
m	11

Code B

Symbol	Codeword
s	0
p	1
a	10
m	11

Decode the binary pattern 0110 using Code B?

- A. spa
- B. sms
- C. Not enough information to decode

Variable length codes

Variable length codes have to necessarily be prefix codes for correct decoding

A prefix code is one where no symbol's codeword is a prefix of another

Code A

Symbol	Codeword
s	00
p	01
a	10
m	11

Code B

Symbol	Codeword
s	0
p	1
a	10
m	11

Code B is not a prefix code

Variable length codes

Is code C better than code A and code B? (Why or why not?)

A. Yes

B. No

Code A

Symbol	Codeword
s	00
p	01
a	10
m	11

Code B

Symbol	Codeword
s	0
p	1
a	10
m	11

Code C

Symbol	Codeword
s	0
p	10
a	110
m	111

Symbol Frequency

s	0.6
p	0.2
a	0.1
m	0.1

Variable length codes

Symbol	Frequency
s	0.6
p	0.2
a	0.1
m	0.1

Code A

Symbol	Codeword
s	00
p	01
a	10
m	11

Code B

Symbol	Codeword
s	0
p	1
a	10
m	11

Code C

Symbol	Codeword
s	0
p	10
a	110
m	111

Average length (code A) = 2 bits/symbol

Average length (code B) = $0.6 * 1 + 0.2 * 1 + 0.1 * 2 + 0.1 * 2$
= 1.2 bits/symbol

Average length (code C) = $0.6 * 1 + 0.2 * 2 + 0.1 * 3 + 0.1 * 3$
= 1.6 bits/symbol

What is the advantage of thinking of codes as trees?

Code A

Symbol	Codeword
s	00
p	01
a	10
m	11

Code B

Symbol	Codeword
s	0
p	1
a	10
m	11

Code C

Symbol	Codeword
s	0
p	10
a	110
m	111

Problem Definition

```
mapsmppam  
ssampamsmam  
...
```

TEXT FILE

Symbol	Frequency
s	0.6
p	0.2
a	0.1
m	0.1

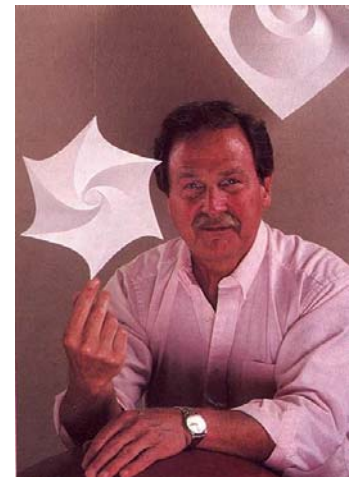
Huffman coding is one of the fundamental ideas that people in computer science and data communications are using all the time - Donald Knuth

Given a frequency distribution over M symbols, find the optimal prefix binary code i.e. one that minimizes the average code length

In short:

I give you frequencies, you give me the best tree!

David Huffman



Problem Definition (reworded for trees)

Input: The frequency (p_i) of occurrence of each symbol (S_i)

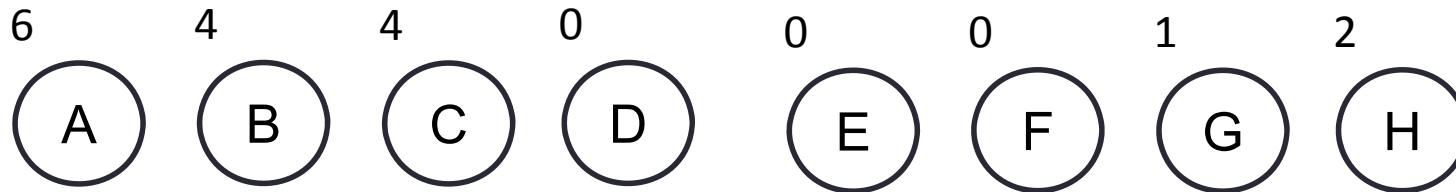
Output: Binary tree T that minimizes the following objective function:

$$L(T) = \sum_{i=1:N} p_i \cdot \text{Depth}(S_i \text{ in } T)$$

Solution: Huffman Codes

Huffman's algorithm: Bottom up construction

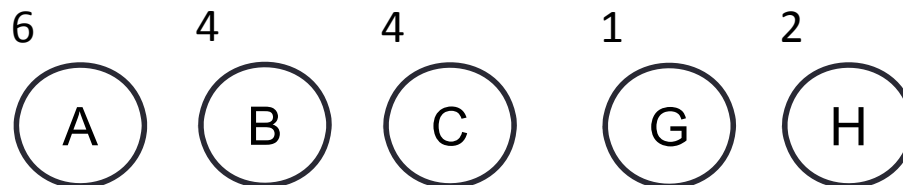
- Build the tree from the bottom up!
- Start with a forest of trees, all with just one node



A: 6; B: 4; C: 4; D: 0; E: 0; F: 0; G: 1; H: 2

Huffman's algorithm: Bottom up construction

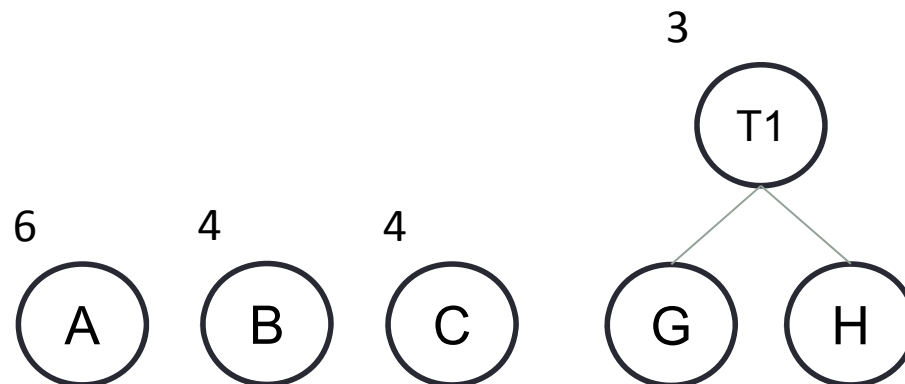
- Build the tree from the bottom up!
- Start with a forest of trees, all with just one node
- Choose the two smallest trees in the forest and merge them



A: 6; B: 4; C: 4; D: 0; E: 0; F: 0; G: 1; H: 2

Huffman's algorithm: Bottom up construction

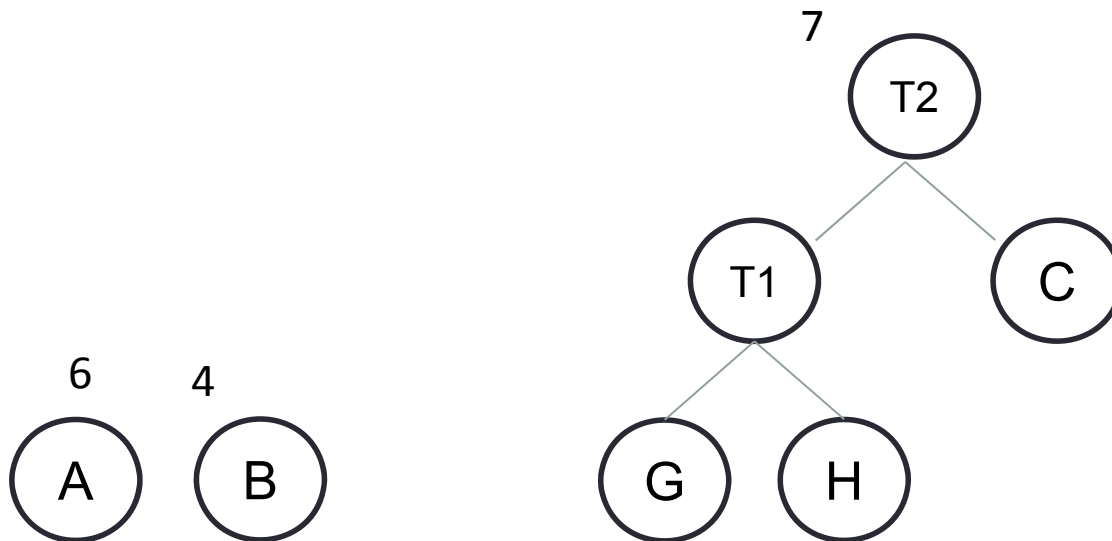
- Build the tree from the bottom up!
- Start with a forest of trees, all with just one node
- Choose the two smallest trees in the forest and merge them



A: 6; B: 4; C: 4; D: 0; E: 0; F: 0; G: 1; H: 2

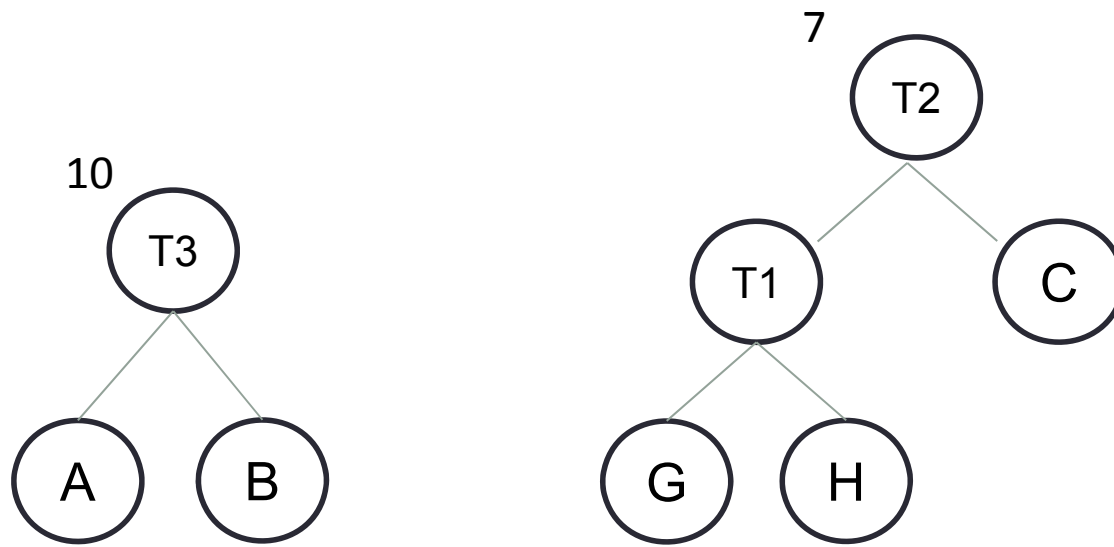
Huffman's algorithm: Bottom up construction

- Build the tree from the bottom up!
- Start with a forest of trees, all with just one node
- Choose the two smallest trees in the forest and merge them
- Repeat until all nodes are in the tree



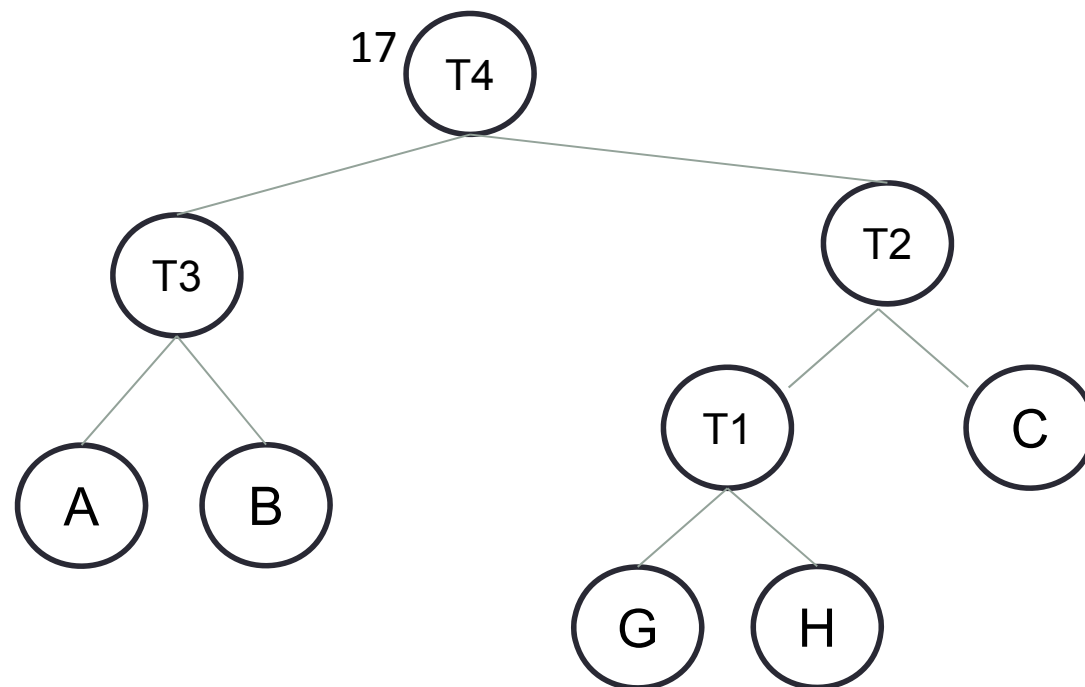
Huffman's algorithm: Bottom up construction

- Build the tree from the bottom up!
- Start with a forest of trees, all with just one node
- Choose the two smallest trees in the forest and merge them
- Repeat until all nodes are in the tree



Huffman's algorithm: Bottom up construction

- Build the tree from the bottom up!
- Start with a forest of trees, all with just one node
- Choose the two smallest trees in the forest and merge them
- Repeat until all nodes are in the tree



PA2: encoding/decoding

ENCODING:

- 1. Scan text file to compute frequencies**
- 2. Build Huffman Tree**
- 3. Encode: Find code for every symbol (letter)**
- 4. Create new compressed file by saving the entire code at the top of the file followed by the code for each symbol (letter) in the file**

DECODING:

- 1. Read the file header (which contains the code) to recreate the tree**
- 2. Decode each letter by reading the file and using the tree**

This is the logical flow of what needs to happen, it need not be the way you develop the code: Brainstorm in class and in section to see how you can design in a modular way