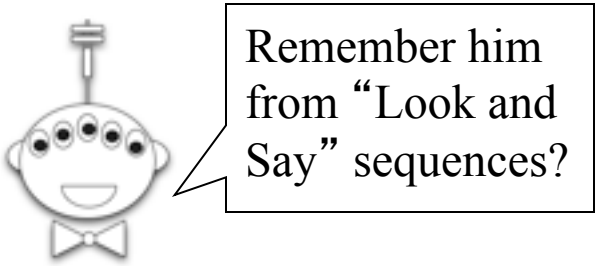
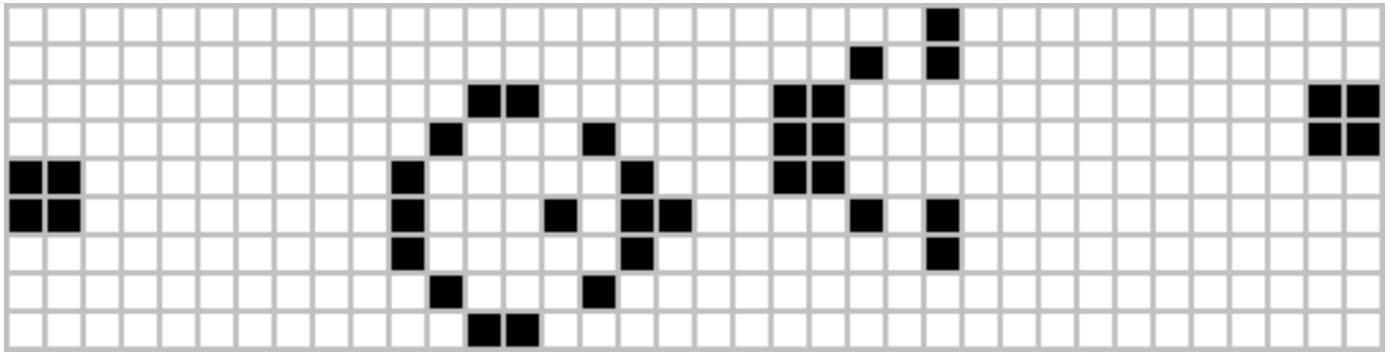
John Conway's Game of Life



John Conway



1. Any live cell with fewer than two neighbours dies of loneliness.
2. Any live cell with more than three neighbours dies of overcrowding.
3. Any live cell with two or three neighbors lives, unchanged, to the next generation.
4. Any dead cell with exactly three neighbors comes to life.

John Conway's Game of Life

Oscillators of varying periods

John Conway's Game of Life

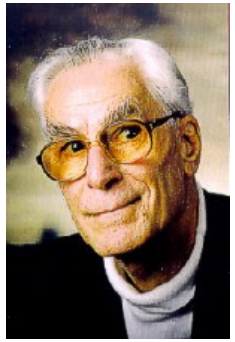
Gosper's Glider "Gun"

Life is “Universal” (1982)

Elwyn Berlekmap

John Conway

Richard Guy



Data Compression

The zzyzva is known to be a xenophobic creature with a zealous personality…

compression algorithm (e.g. zip)

B6^9)=\n%
%spam!
=&&penguin/
?’ ,/+

Now we can delete the original file!

TEXT FILE
zzyzva.txt

58,254 bytes

TEXT FILE
zzyzva.txt.Z

23,124 bytes

Data Compression!

The zzyzva is known to be a xenophobic creature with a zealous personality…

TEXT FILE

Letter	ord(Letter)	Binary
T	84	01010100
h	104	01101000
e	101	01100101
z	122	01111010

- " "- 1226754 19.04%
- E - 655257 10.17%
- T - 474521 7.37%
- A - 425718 6.61%
- … skipping a few …
- J - 5329 0.08%
- Q - 4923 0.08%
- Z - 3378 0.05%

English text letter frequencies

But these statistics are on average, not for my essay on the zzyzva!

The Prefix Property

The zzyzva is known to be a xenophobic creature with a zealous personality…

TEXT FILE

Letter	frequency	Binary code
z	0.25	00
y	0.10	01
x	0.09	10
a	0.08	111
r	0.02	1100

101110100001100 = 10 111 01 00 00 1100

The Variable Length Coding Problem…

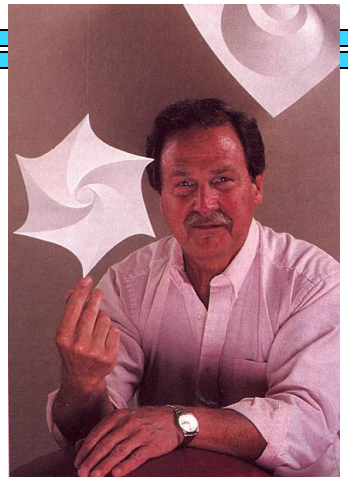
Letter	Frequency
a_1	$freq(a_1)$
a_2	$freq(a_2)$
a_3	$freq(a_3)$
…	
a_n	$freq(a_n)$

These frequencies are from the specific file that we're planning to compress!!

Objective: Find a binary prefix code that minimizes…

$freq(a_1) \times codelength(a_1) +$
$freq(a_2) \times codelength(a_2) + \ldots$
$freq(a_n) \times codelength(a_n)$

The David Huffman Story!



map smppam
ssampamsmam
…

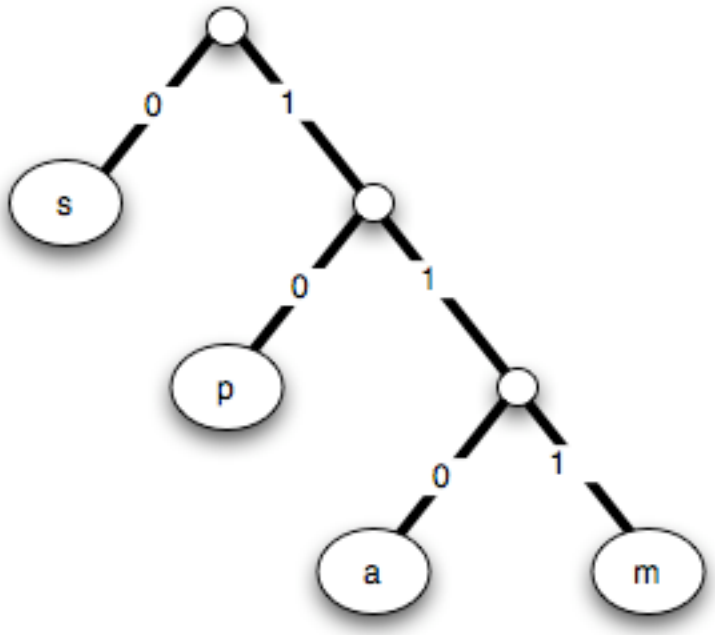
TEXT FILE

Letter	freq
s	0.6
p	0.2
a	0.1
m	0.1

ENCODING:

1. **Scan text file to compute frequencies**
2. **Build Huffman Tree**
3. **Find code for every symbol (letter) - why is this a prefix code?**
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**

I wonder about trees - Robert Frost
We wonder about Robert Frost - Trees

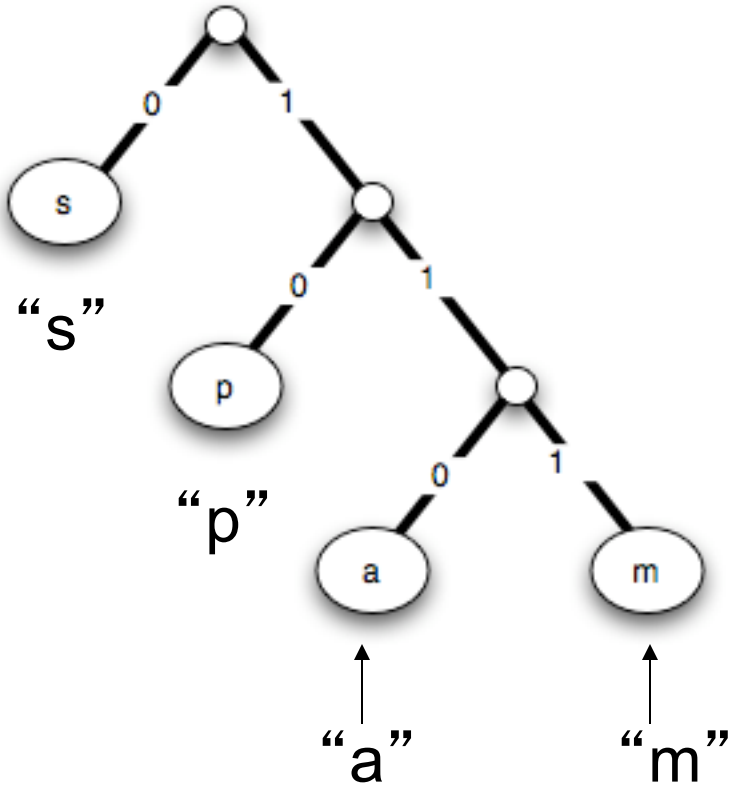


Recursive definition of a tree…

A tree is:

1. Just a symbol (I.e. a “leaf”) or
2. A left subtree and a right subtree

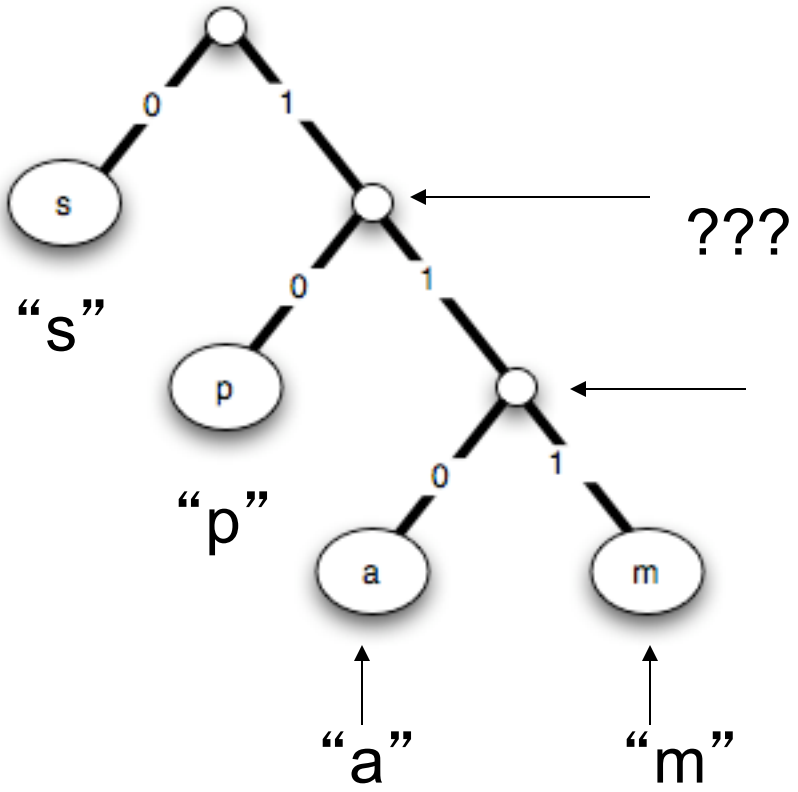
I wonder about trees - Robert Frost
We wonder about Robert Frost - Trees



Recursive definition of a tree…
A tree is:

1. Just a symbol (i.e. a "leaf") or
2. A left subtree and a right subtree

I wonder about trees - Robert Frost
We wonder about Robert Frost - Trees

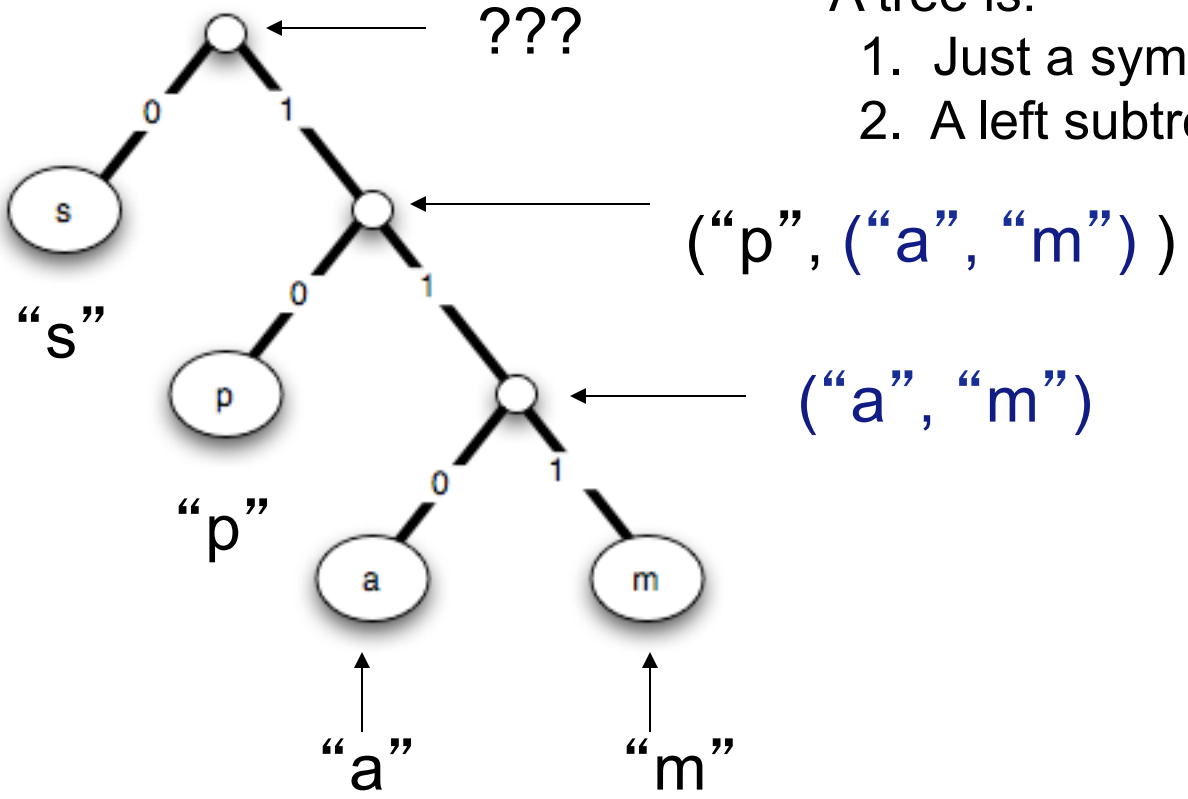


Recursive definition of a tree…
A tree is:

1. Just a symbol (i.e. a "leaf") or
2. A left subtree and a right subtree

Shouldn't that be the list ["a", "m"] ?

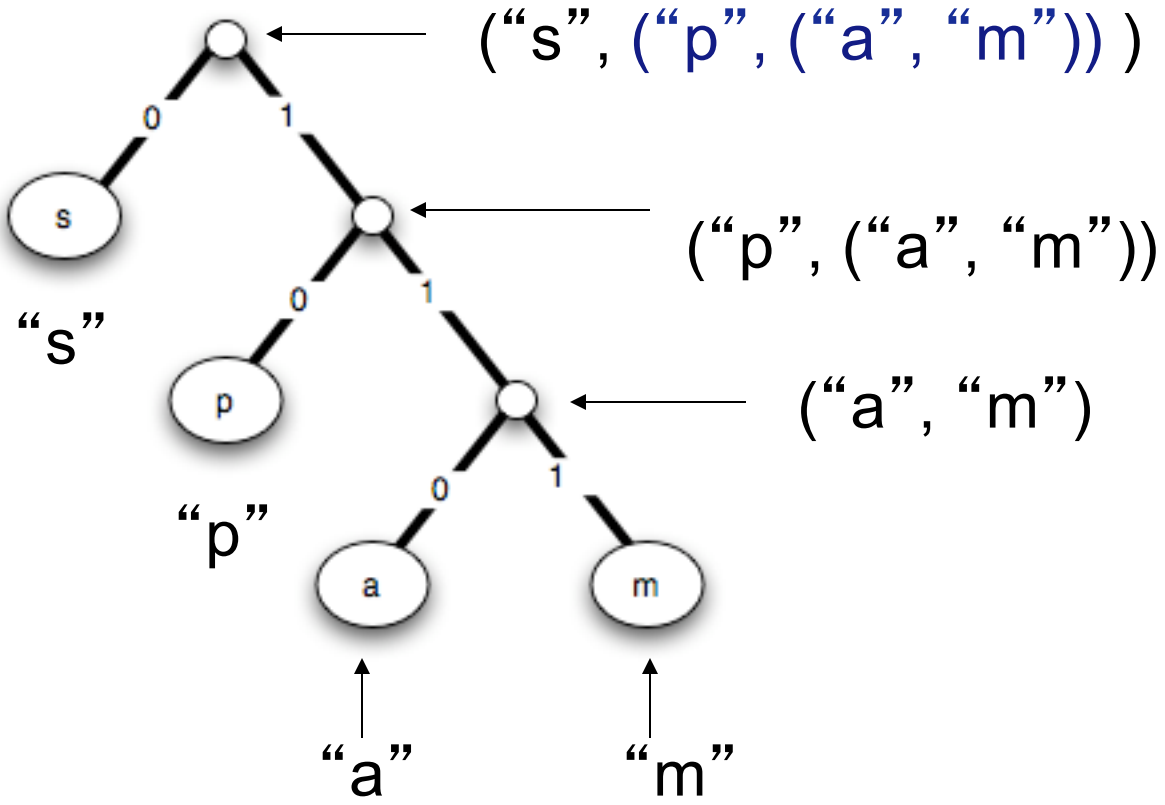
I wonder about trees - Robert Frost
We wonder about Robert Frost - Trees



Recursive definition of a tree…
A tree is:

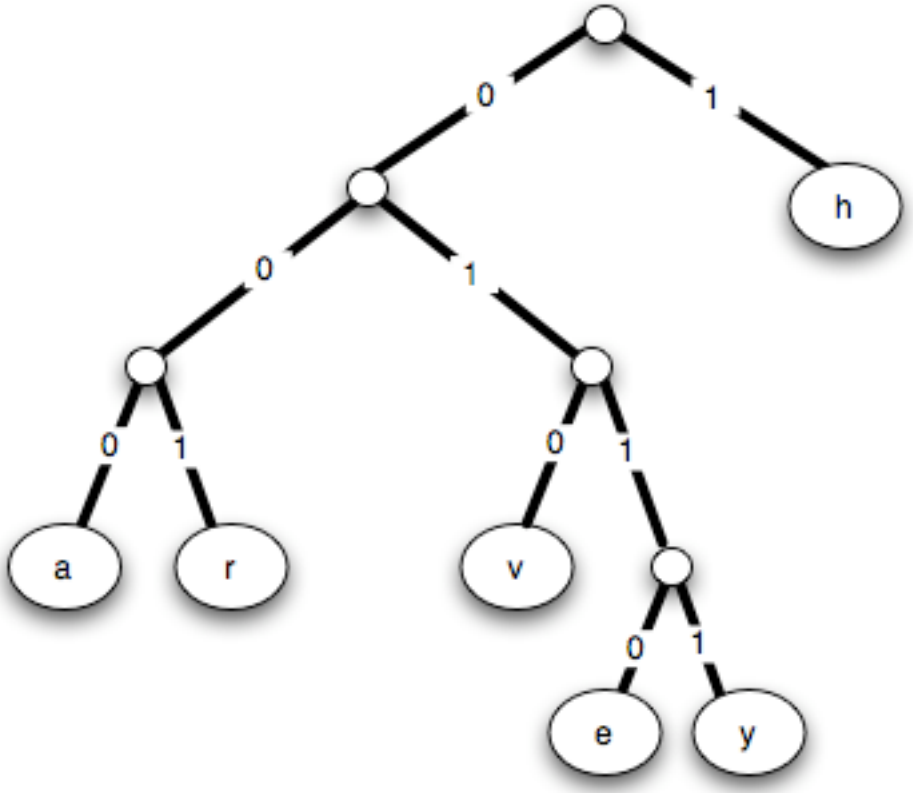
1. Just a symbol (I.e. a "leaf") or
2. A left subtree and a right subtree

I wonder about trees - Robert Frost
We wonder about Robert Frost - Trees

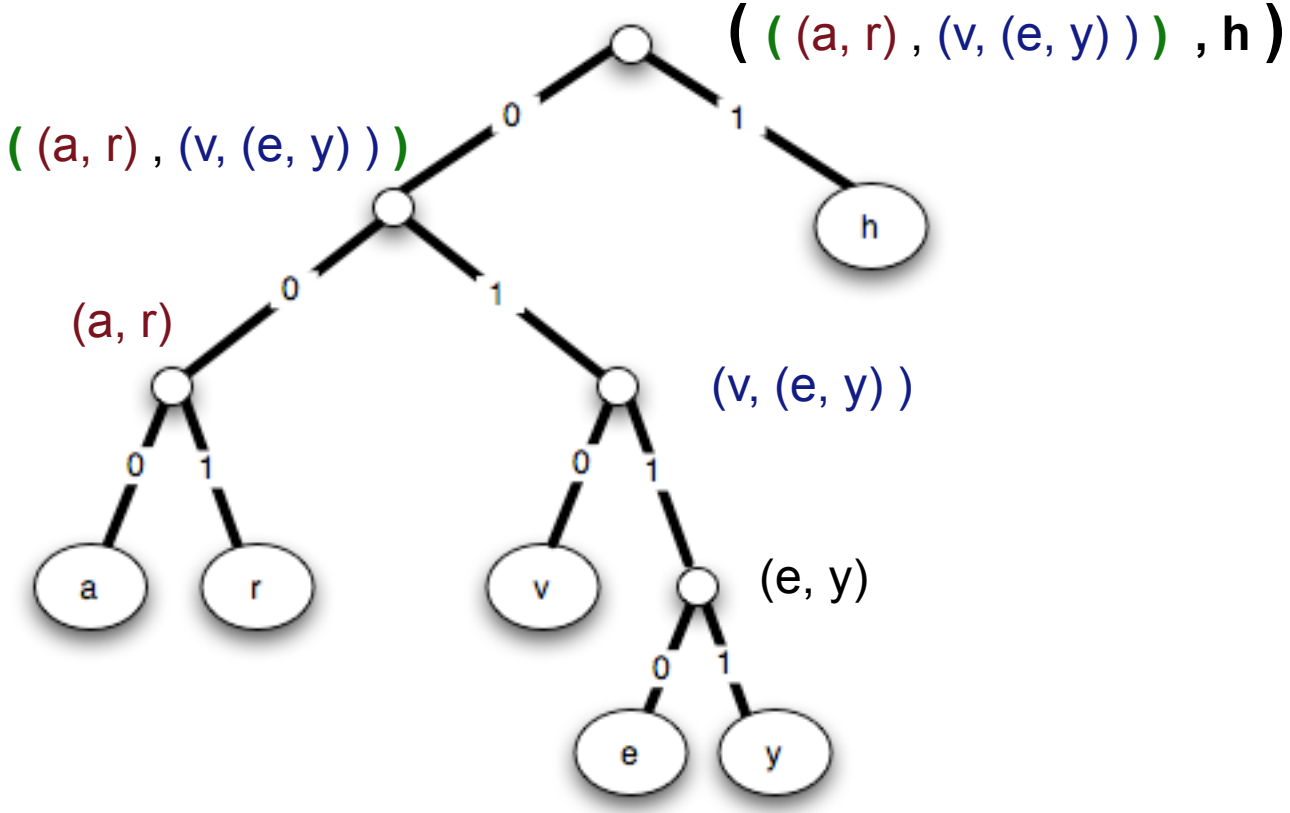


You Try It!

Worksheet



You Try It!



File I/0

```
def read_file():
    filename = raw_input("Enter the name of a file: ")
    myfile = open(filename, "r")  # "r" means "read"
    contents = myfile.read()
    myfile.close()
    return contents

def write_file(string):
    filename = raw_input("Enter the name of a file: ")
    myfile = open(filename, "w")  # "w" means "write"
    myfile.write(string)
    myfile.close()
    return
```

Careful NOT to call this "file"
(`file` is a built-in name in python)

```
>>> write_file("Spam is the secret to all :^)!")
Enter the name of a file to write: spam.txt
>>> read_file()
Enter the name of a file: spam.txt
'Spam is the secret to all :^)!'
```

What about line breaks in the file?