

## Source Coding

by

**Dr. Manas Khatua**

Assistant Professor

Dept. of CSE

IIT Jodhpur

E-mail: [manaskhatua@iitj.ac.in](mailto:manaskhatua@iitj.ac.in)

Web: <http://home.iitj.ac.in/~manaskhatua>

<http://manaskhatua.github.io/>

# Coding Theory



- Coding theory is the study of the **properties of codes** and their respective **fitness** for specific applications.
- Codes are **used for**
  - Data compression
  - Error-detection and error-correction
  - Networking
  - Cryptography
- the **purpose of coding** is of designing efficient and reliable data transmission methods.
- There are four **types of coding**:
  - **Source coding**
  - Channel coding
  - Line coding
  - Cryptographic coding

# Cont...



- **Source coding**
  - The aim of source coding is to take the source data and **make it smaller in size**.
  - e.g., Zip coding, **Huffman coding**
- **Channel coding**
  - The purpose is to find codes which transmit quickly, contain many valid code words and can **correct or at least detect many errors**.
  - e.g., Reed-Solomon code, Turbo code, LDPC code, Cyclic code, Convolution code
- **Line coding**
  - is called digital **baseband modulation** technique
  - e.g., unipolar, polar, bipolar, and Manchester encoding
- **Cryptographic coding**
  - is the practice and study of techniques for **secure communication** in the presence of third parties
  - e.g., RSA Algorithm

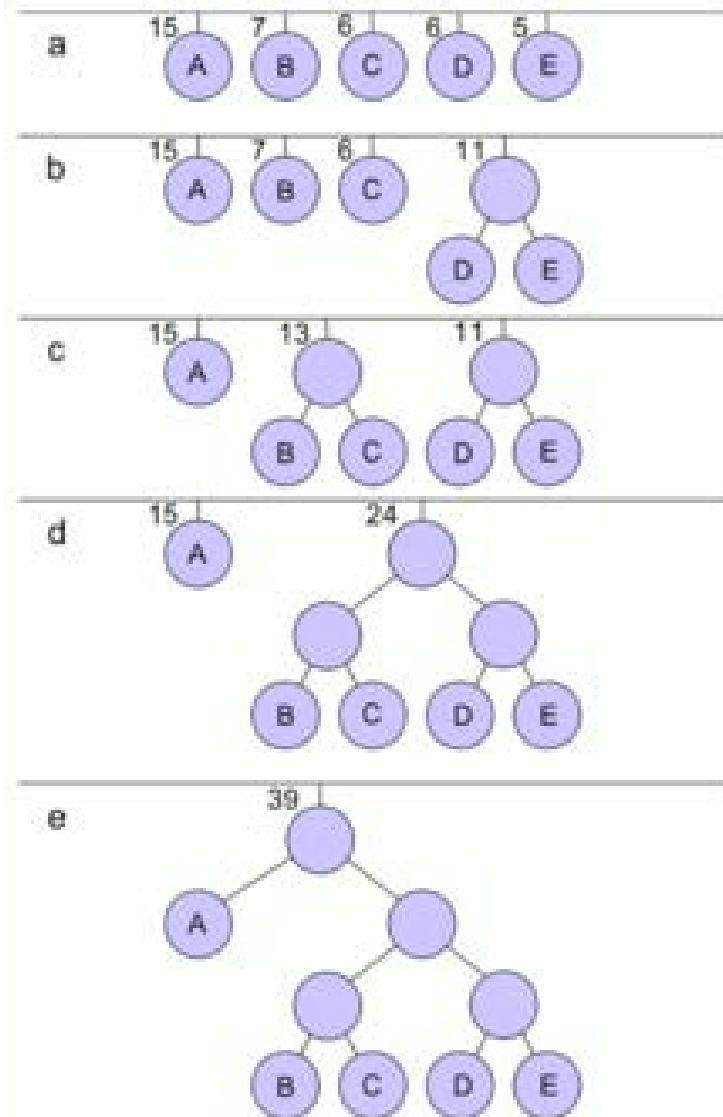
# Huffman Code



- It is commonly used for **lossless data compression**.
- Lossless compression allows the original data to be **perfectly reconstructed** from the compressed data
- Huffman coding **uses a specific method** for choosing the **representation for each symbol**, resulting in a **prefix code**
- Huffman tree is a specific binary tree
- **Huffman Code**: prefix code representation of each symbol in the Huffman tree
- Building the **Huffman Tree** from the **bottom up** guaranteed optimality
- It generates minimum length code

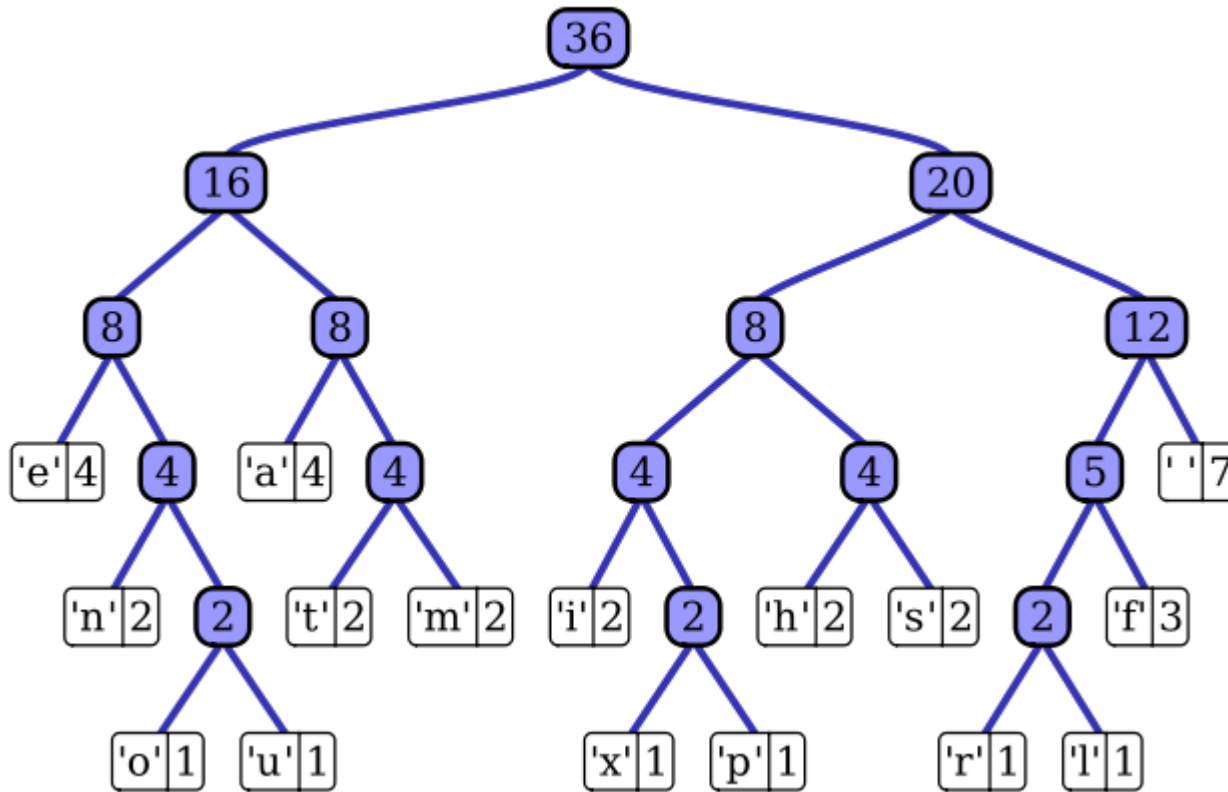
# Huffman Tree

- 15,7,6,6,5
- 15,7,6,(11)
- 15,11,(13)
- 15,(24)
- 39



# Huffman Tree

"this is an example of a huffman tree"



Char	Freq	Code
space	7	111
a	4	010
e	4	000
f	3	1101
h	2	1010
i	2	1000
m	2	0111
n	2	0010
s	2	1011
t	2	0110
l	1	11001
o	1	00110
p	1	10011
r	1	11000
u	1	00111
x	1	10010

# Prefix codes



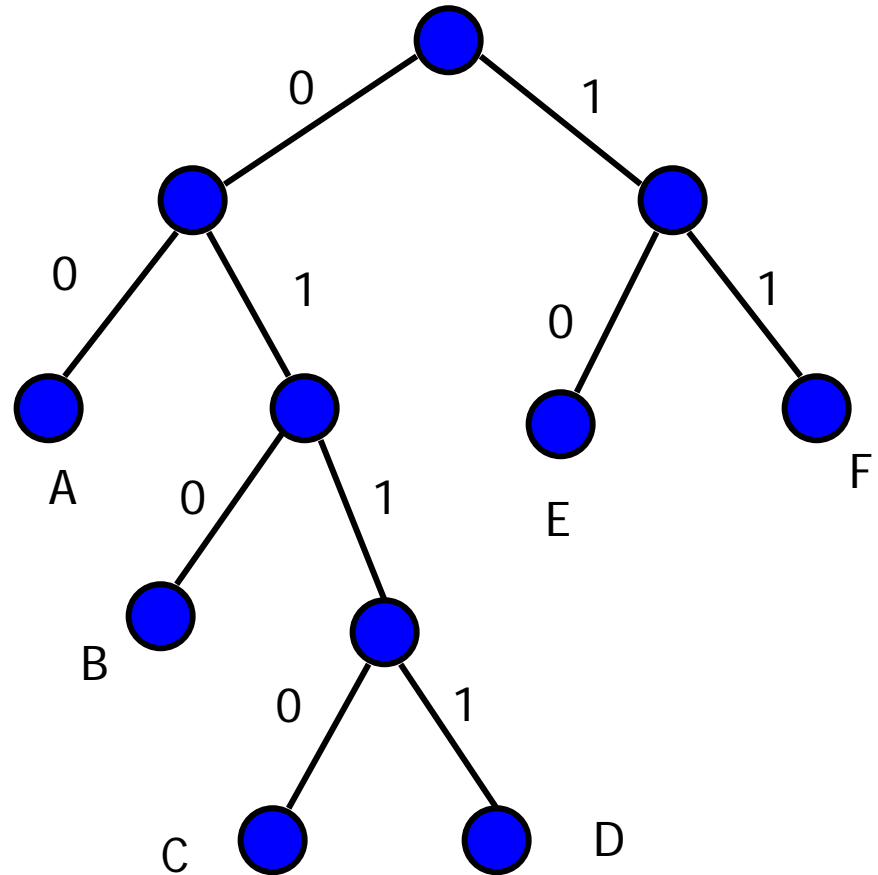
- Prefix(-free) codes
- No prefix of a codeword is a codeword
- Uniquely decodable
- Variable length code

A	00	1	00
B	010	01	10
C	011	001	11
D	100	0001	0001
E	11	00001	11000
F	101	000001	101

# Prefix codes and binary trees

- Tree representation of prefix codes

A	00
B	010
C	0110
D	0111
E	10
F	11



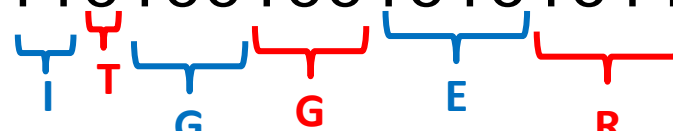


# Uniquely decodable code

- no two distinct strings have the same encoding
- Let  $x$  and  $y$  are two strings.  $x \neq y \Rightarrow c(x) \neq c(y)$

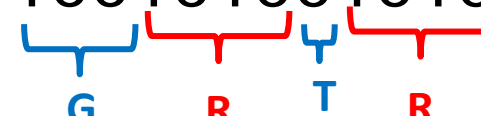
T	0
I	11
G	100
E	1010
R	1011

11010010010101011




T	0
I	10
G	100
E	0111
R	1010

100101001010



100101001010



# Minimum length code

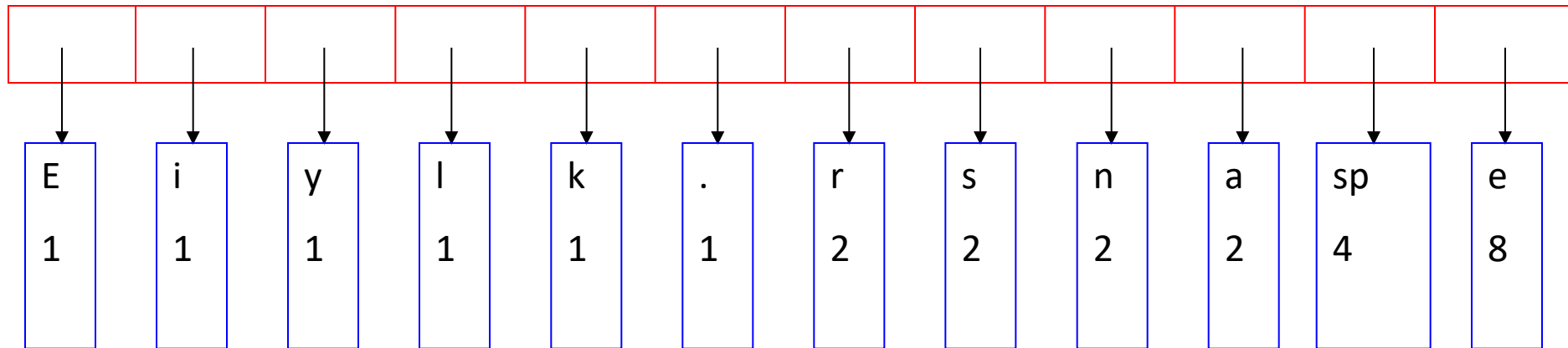
- How to code so that average bits/character is minimized?

	Probability	Code
A	1/4	01
B	1/8	001
C	1/16	0000
D	1/16	0001
E	1/2	1

- Create Huffman tree – prefix codes tree with **minimum weighted path length**
- Derivation Rules**
  - Two rarest items will have the longest codewords
  - Codewords for rarest items differ only in the last bit

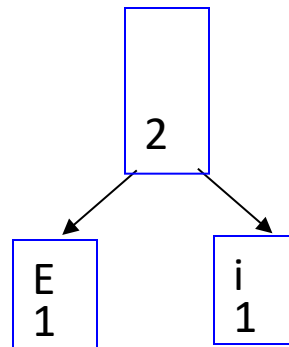
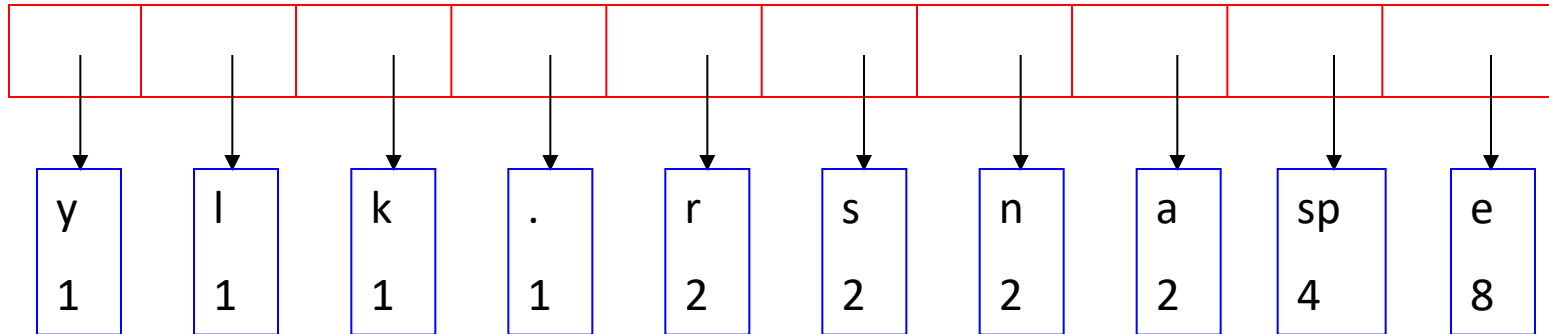
# Huffman Tree using Queue

- The queue after inserting all nodes

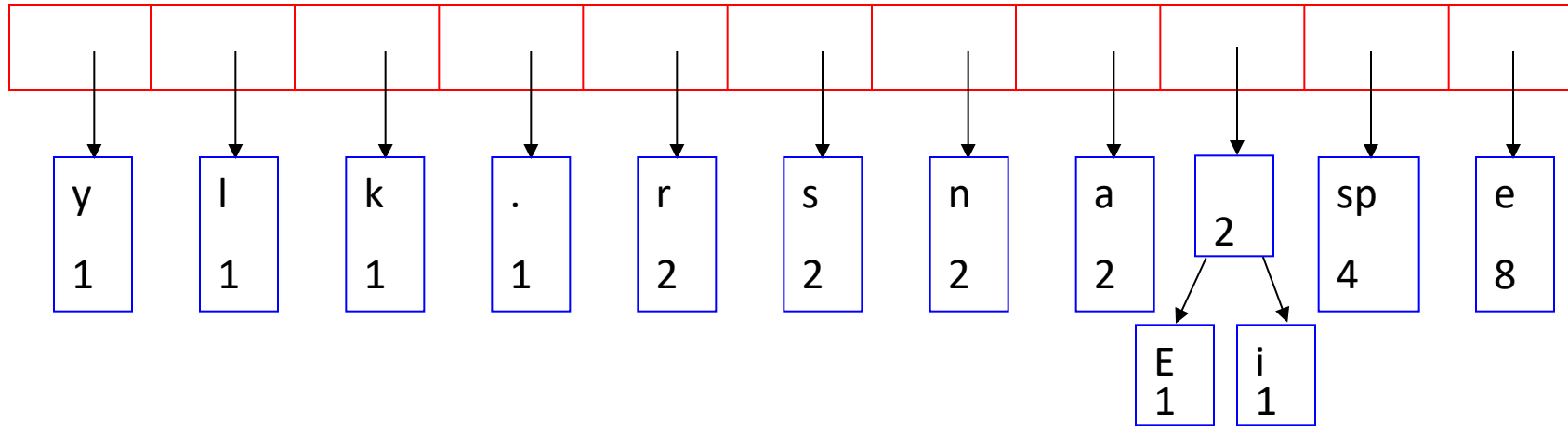


- Null Pointers are not shown

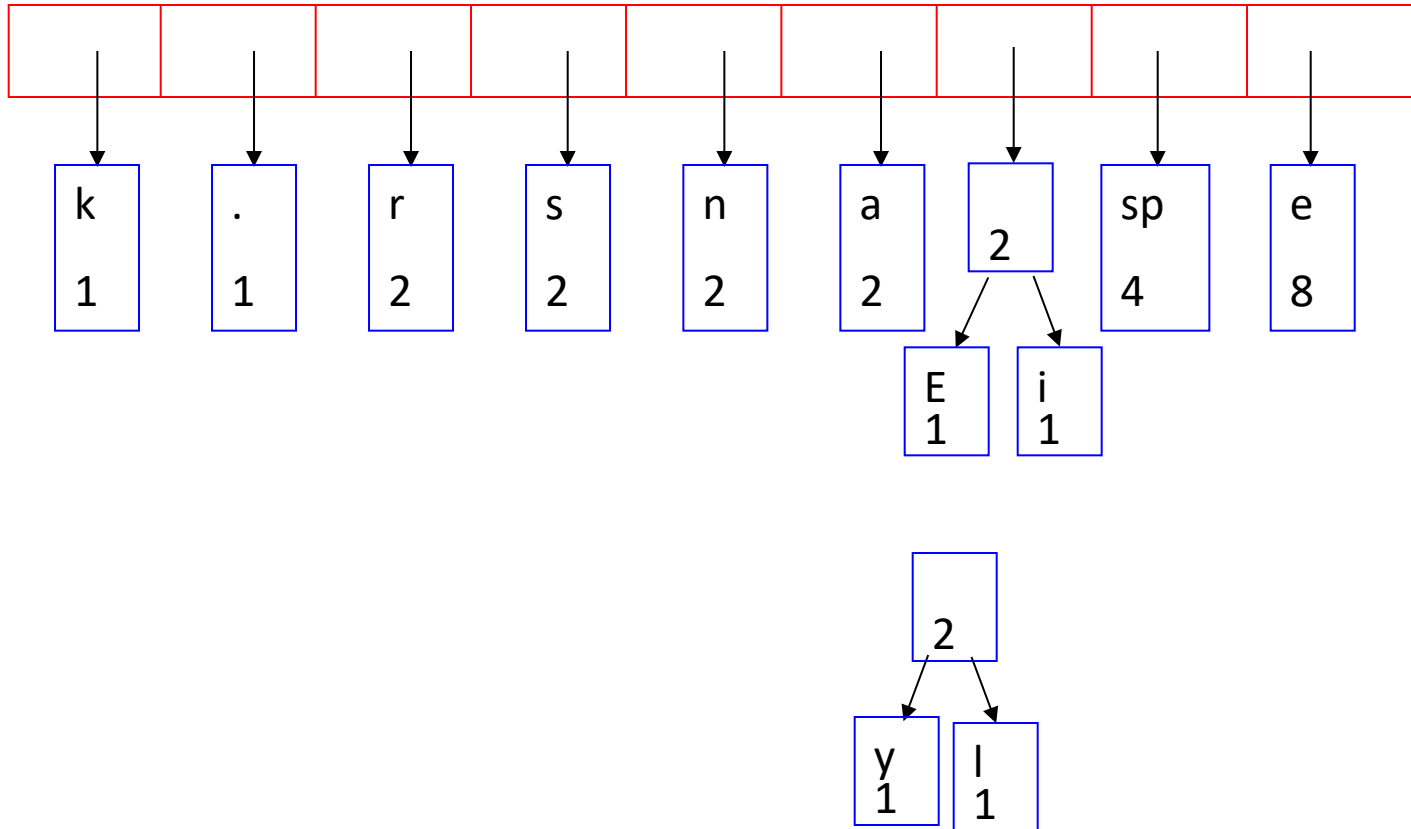
# Cont...



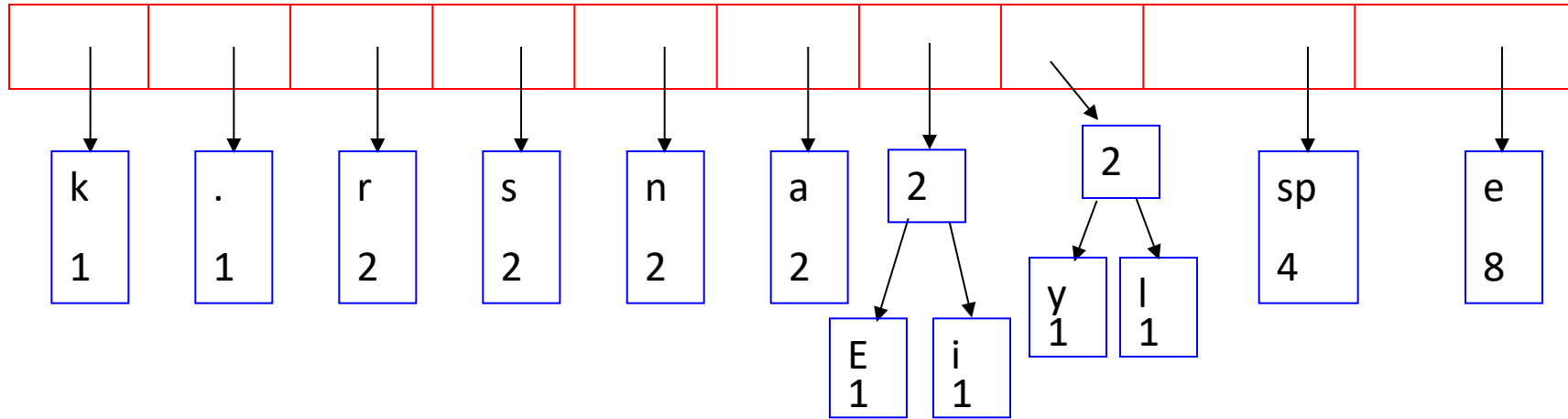
# Cont...



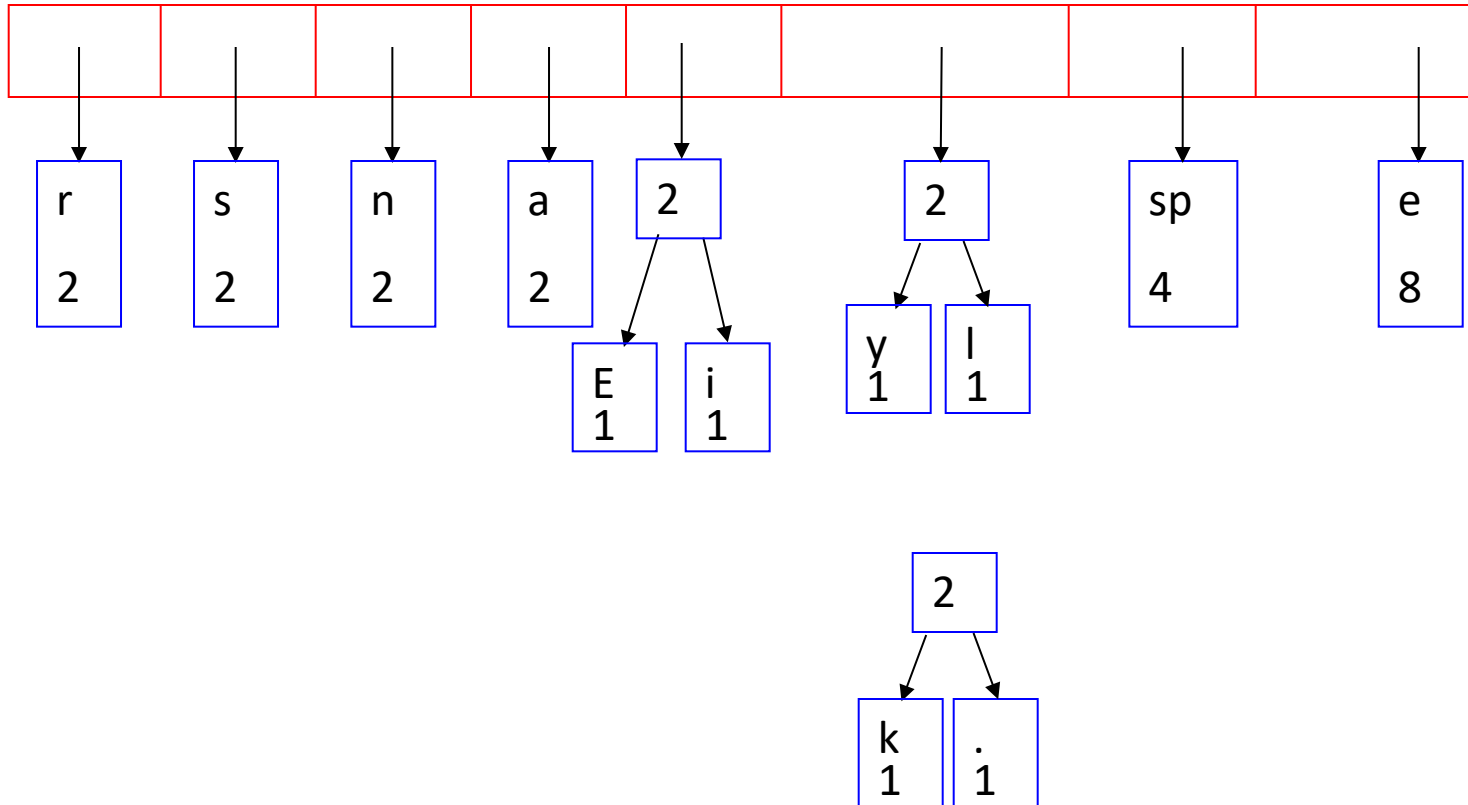
# Cont...



# Cont...

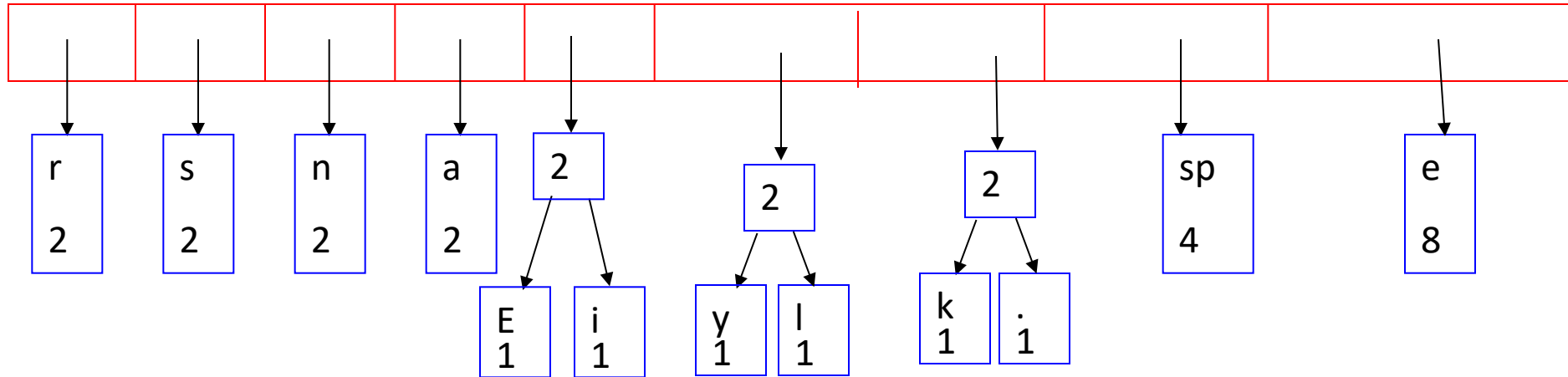


# Cont...

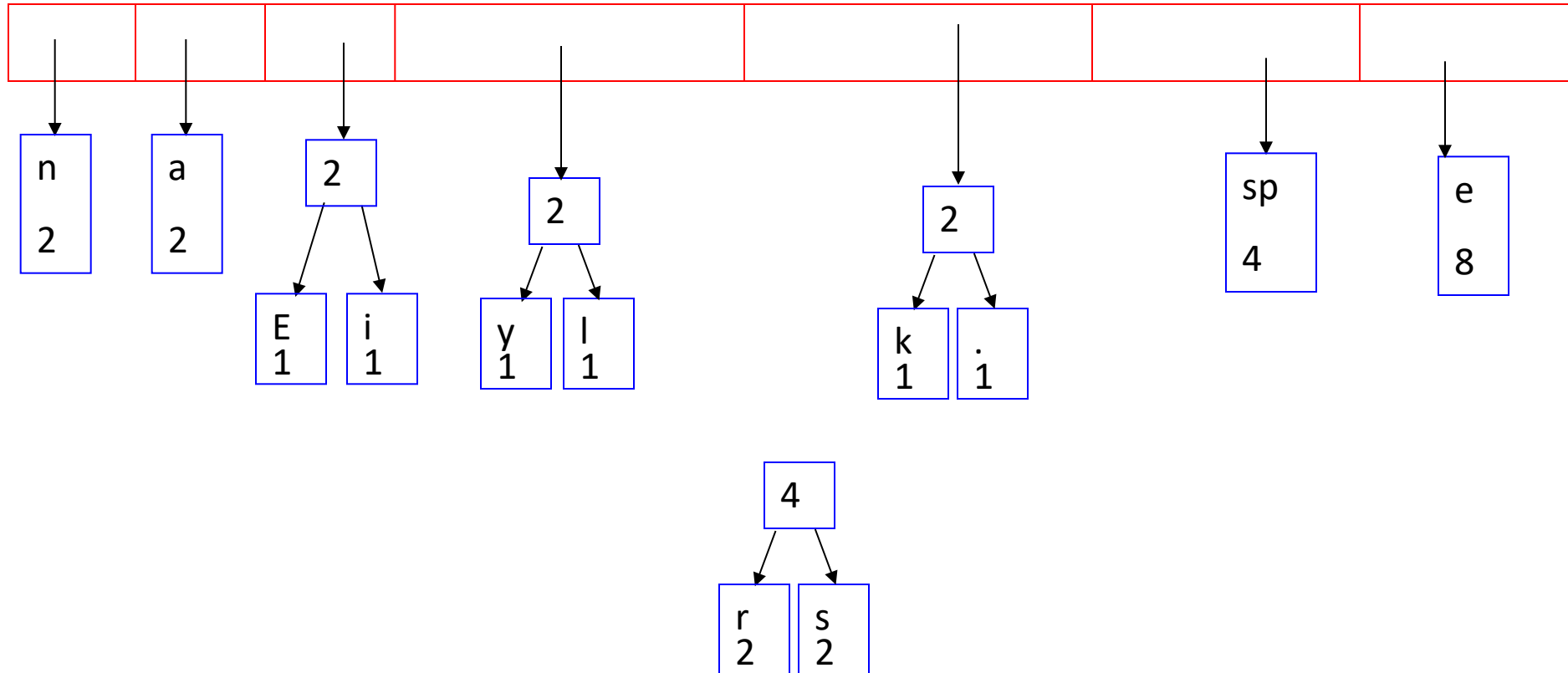




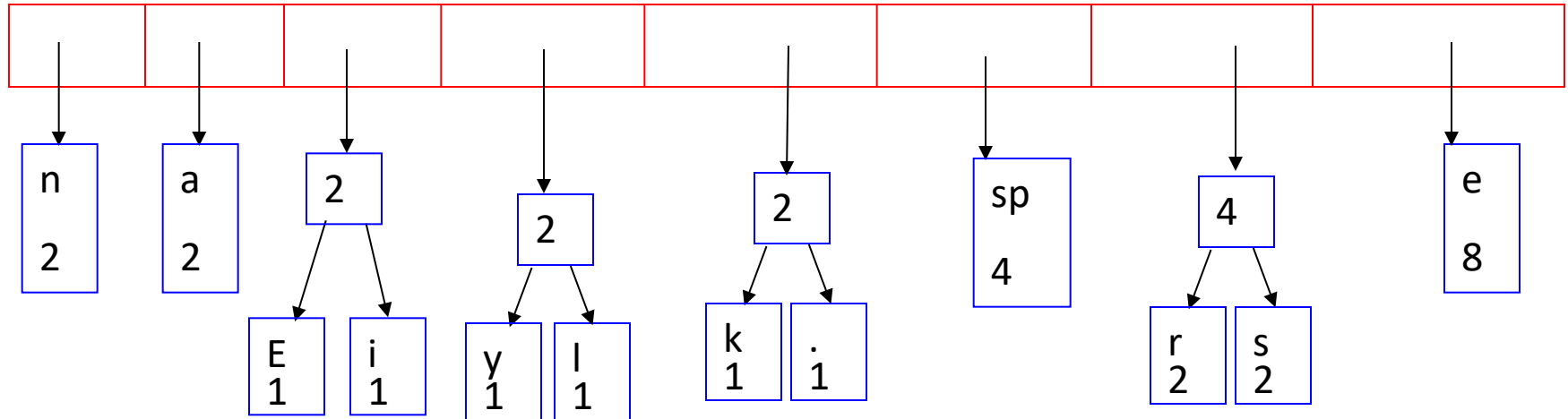
# Cont...



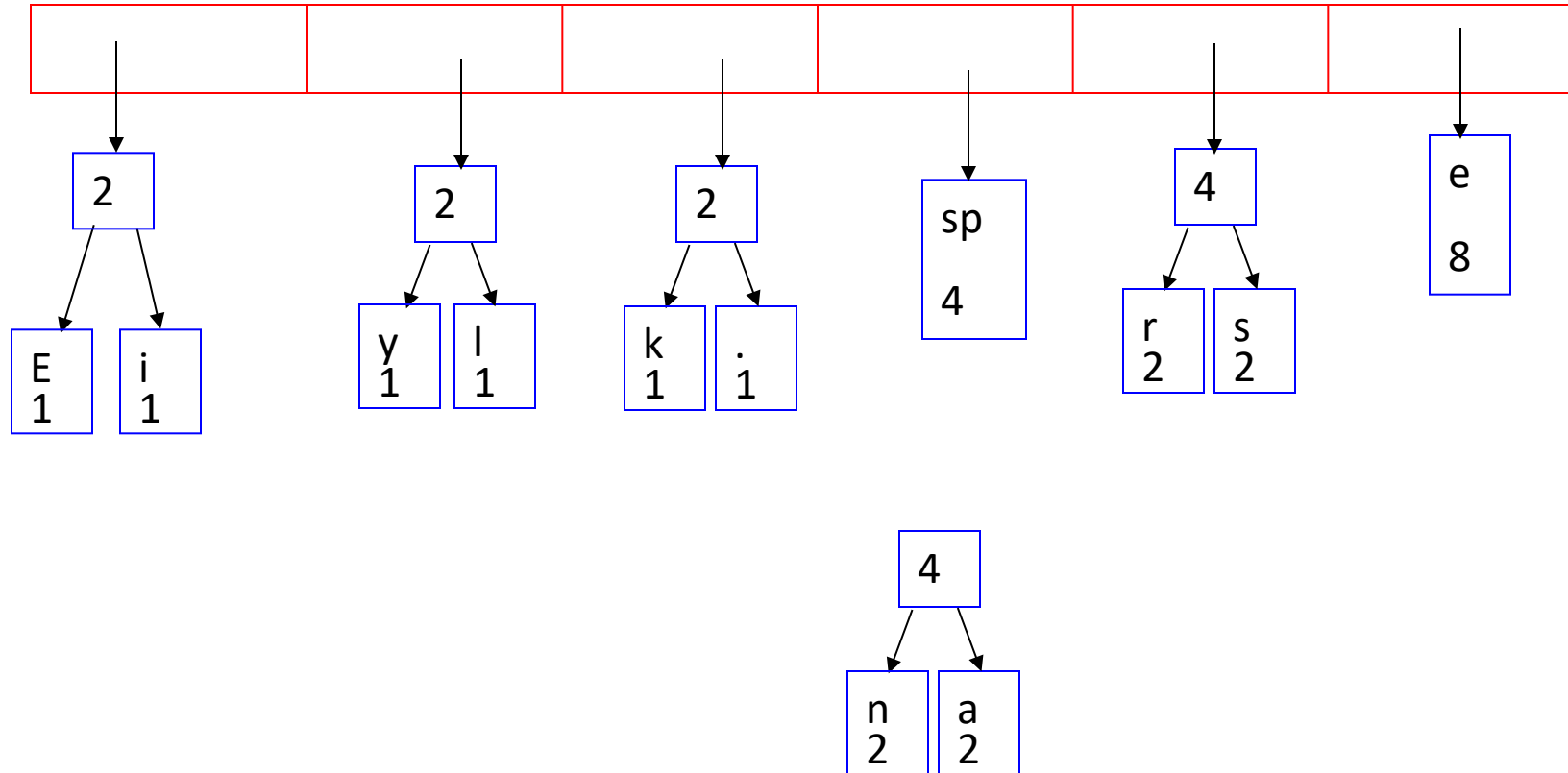
# Cont...



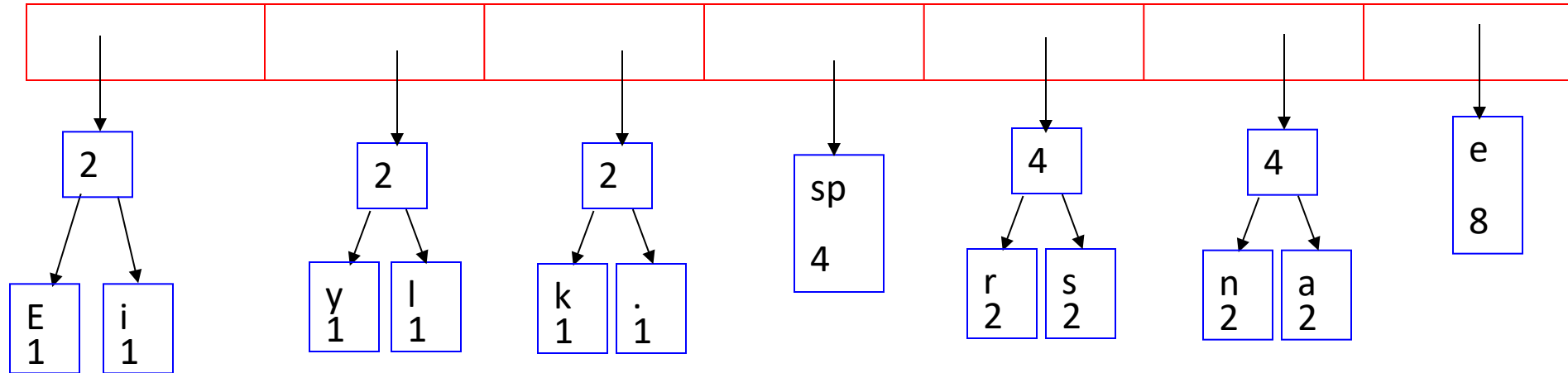
# Cont...



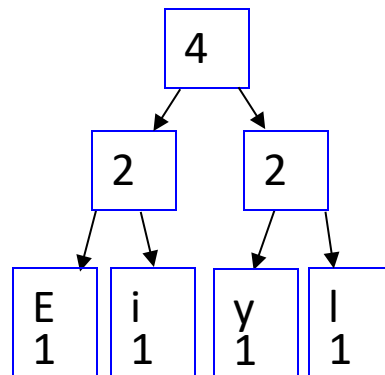
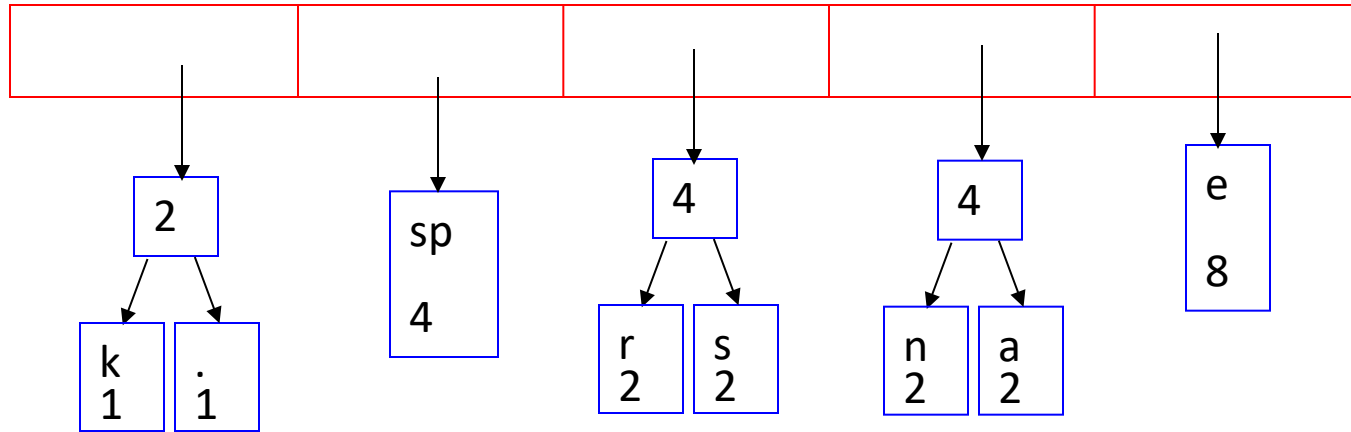
# Cont...



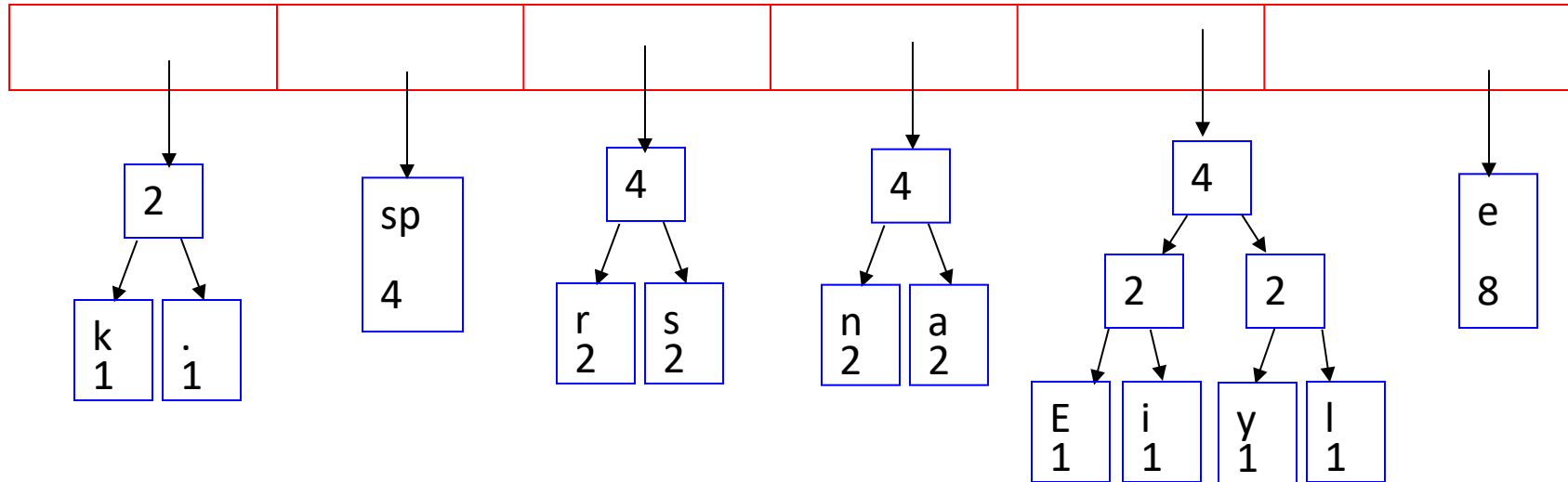
# Cont...



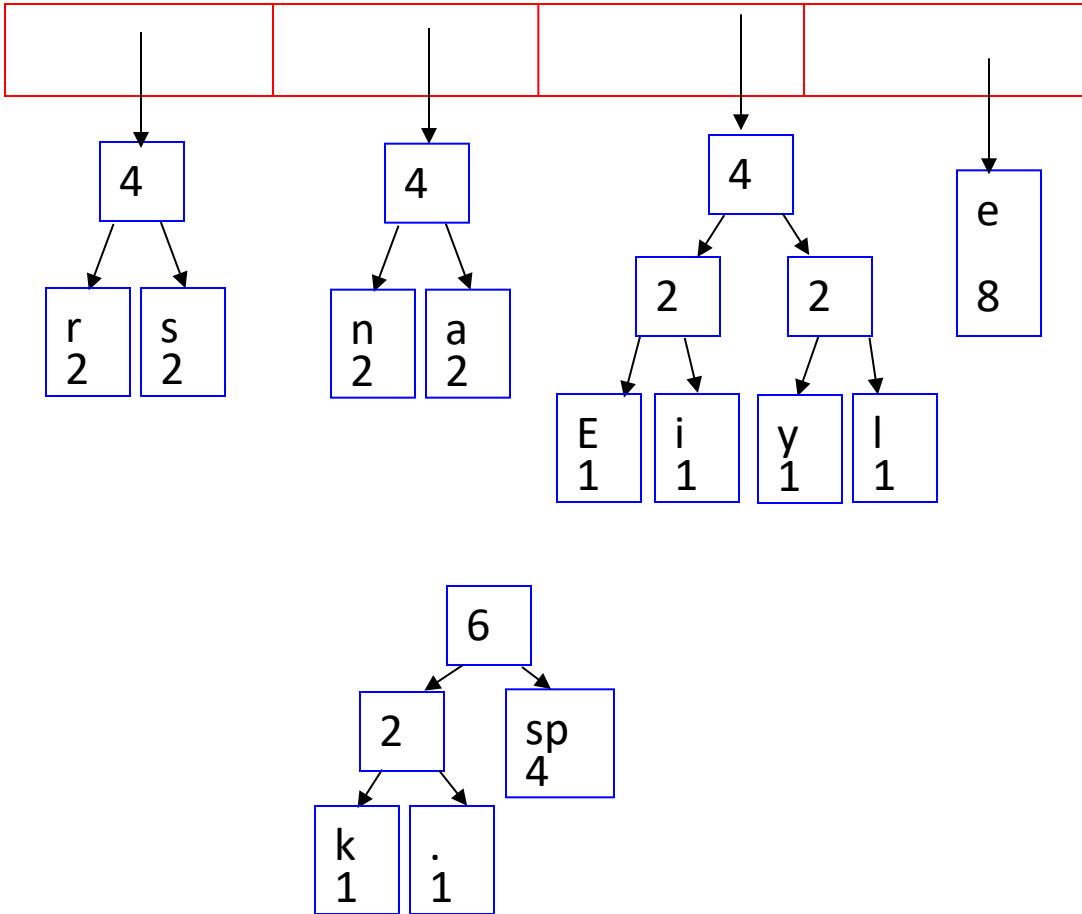
# Cont...



# Cont...

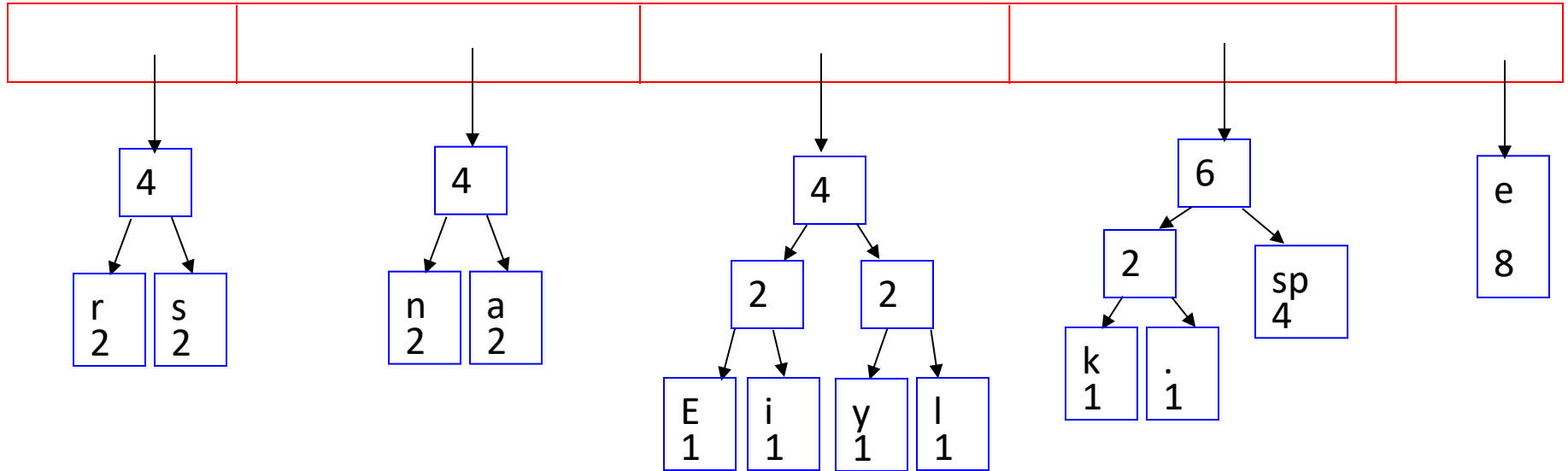


# Cont...

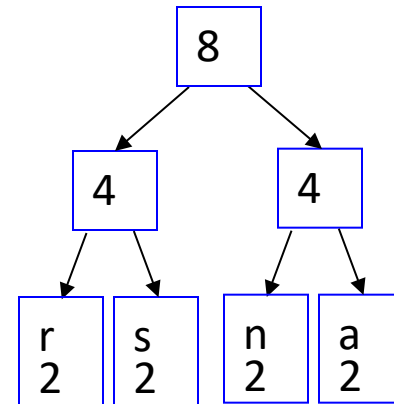
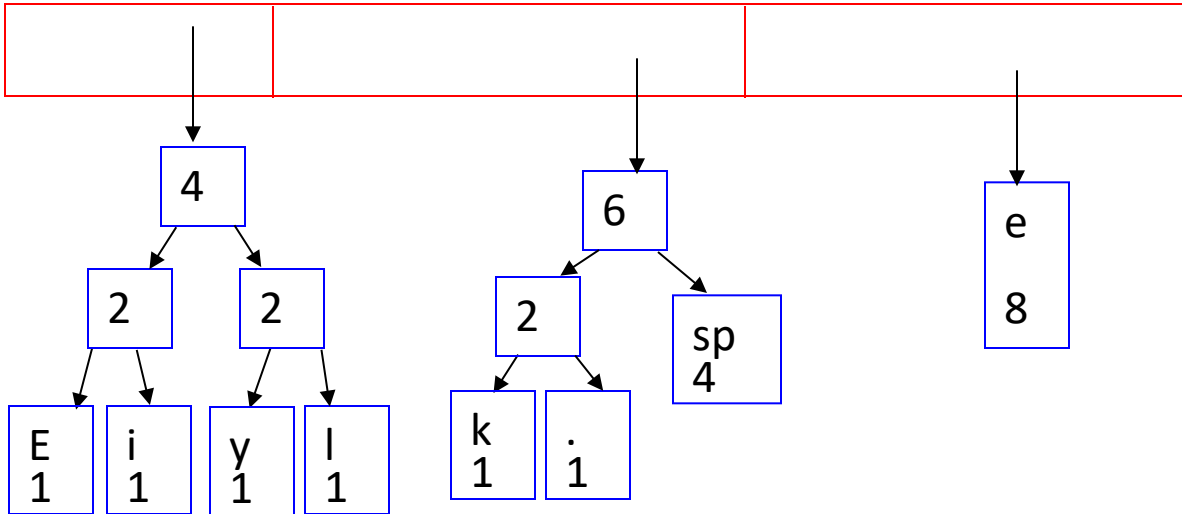




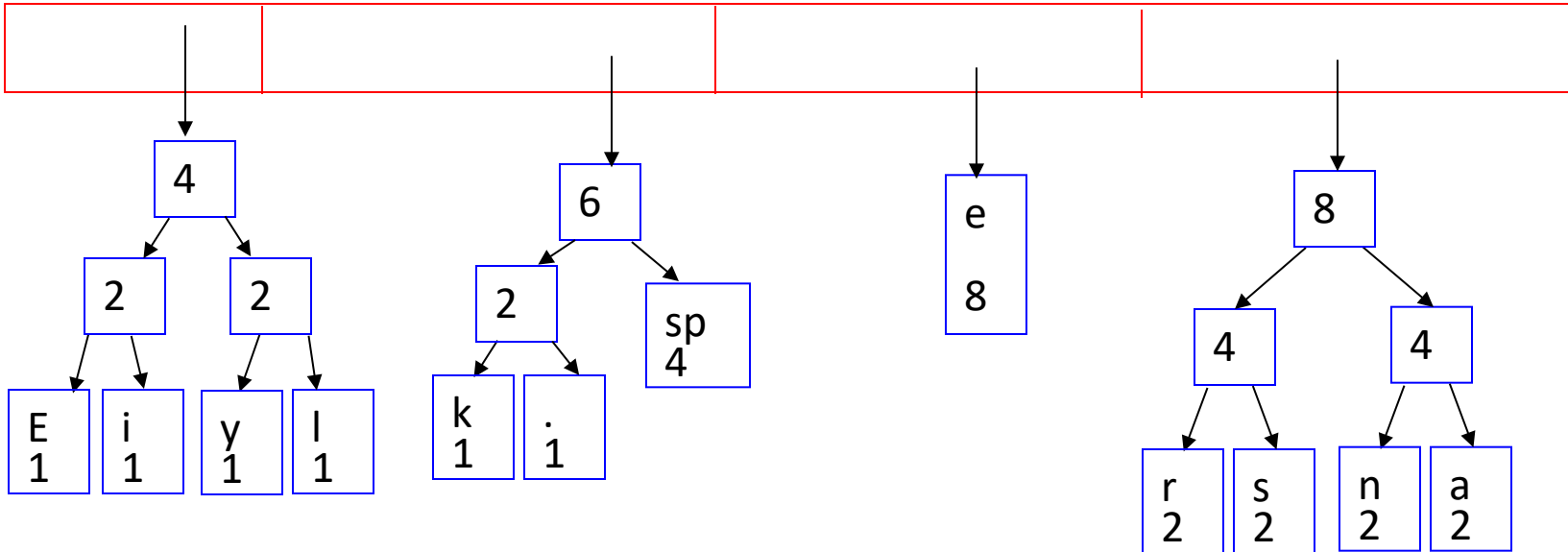
# Cont...



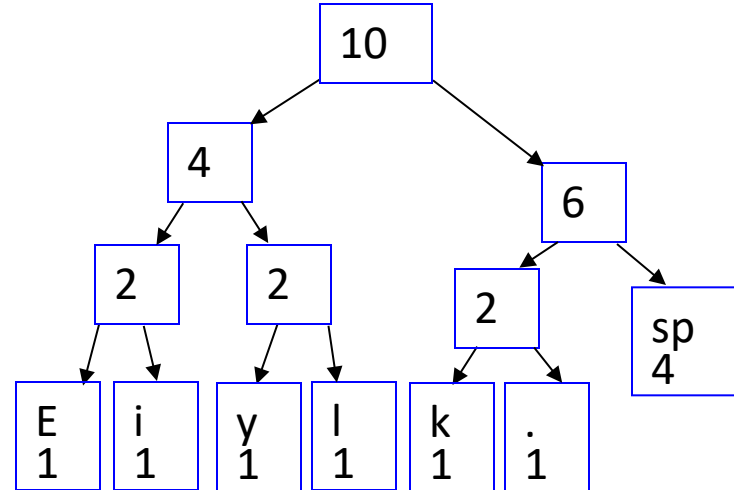
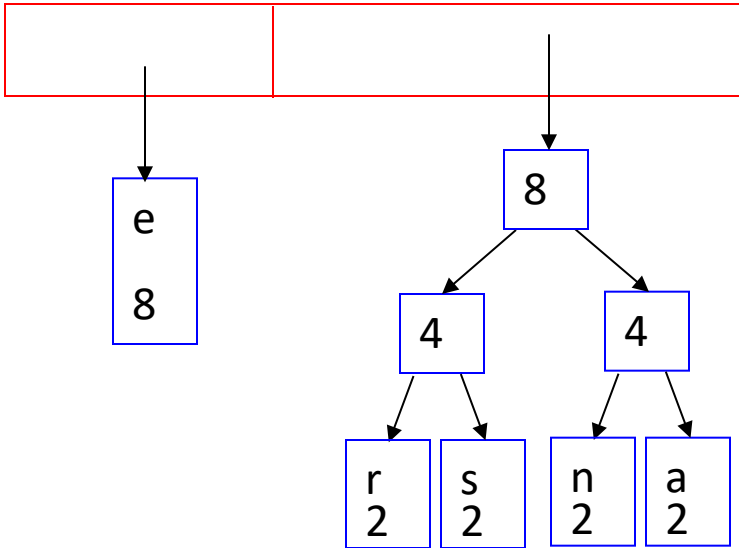
# Cont...



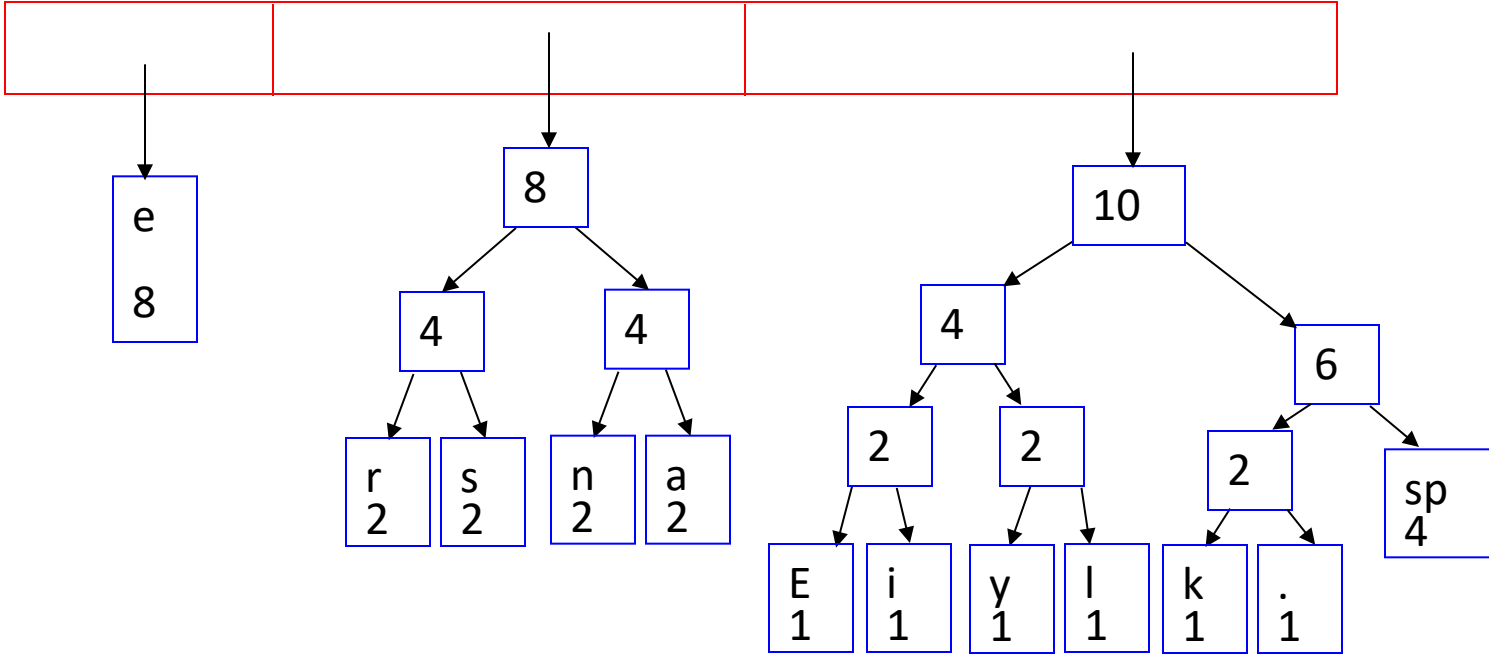
# Cont...



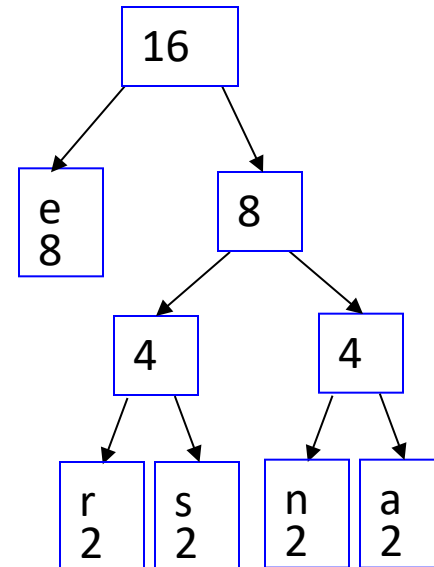
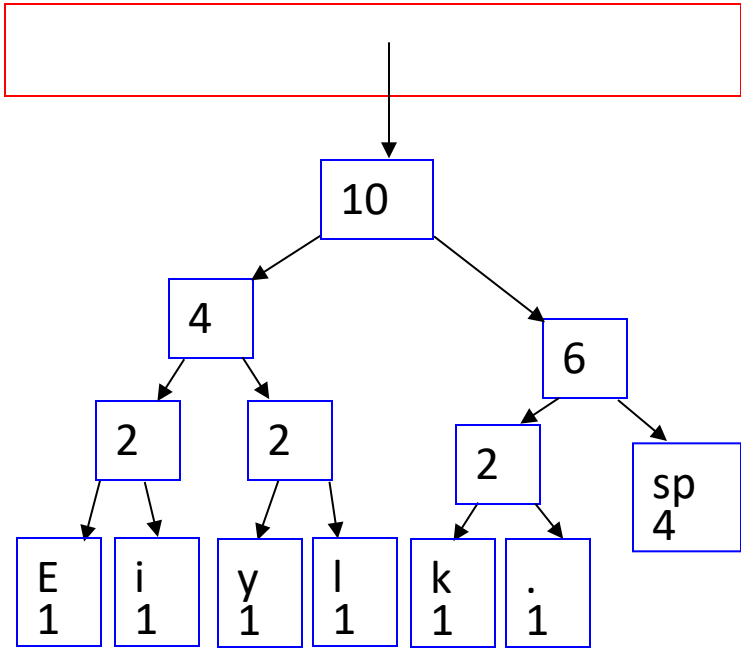
# Cont...



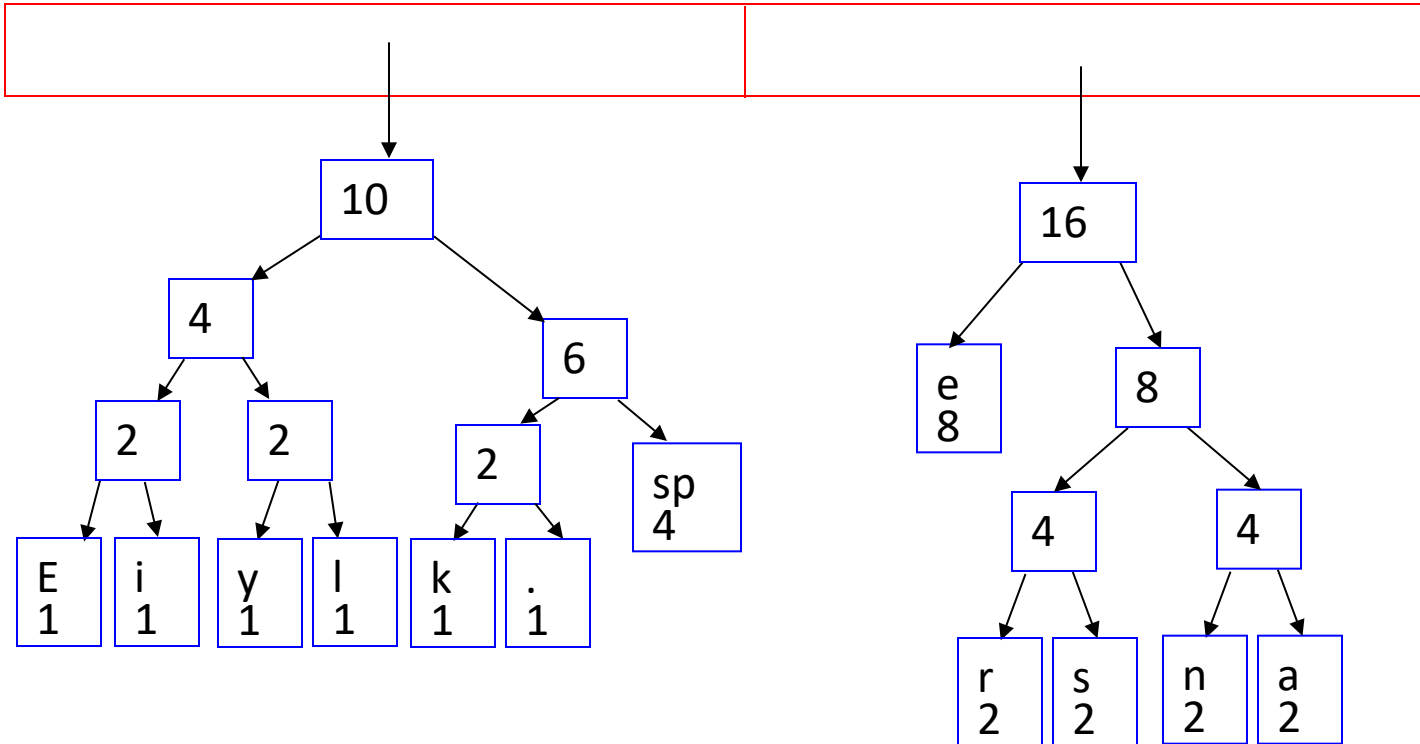
# Cont...



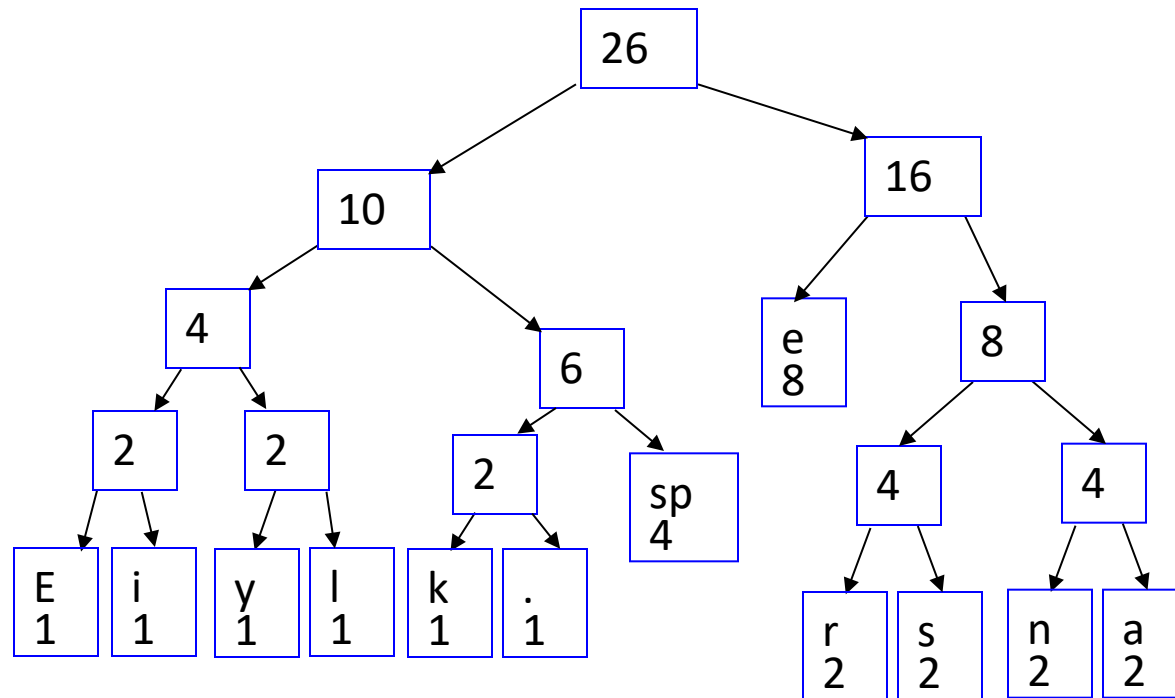
# Cont...



# Cont...

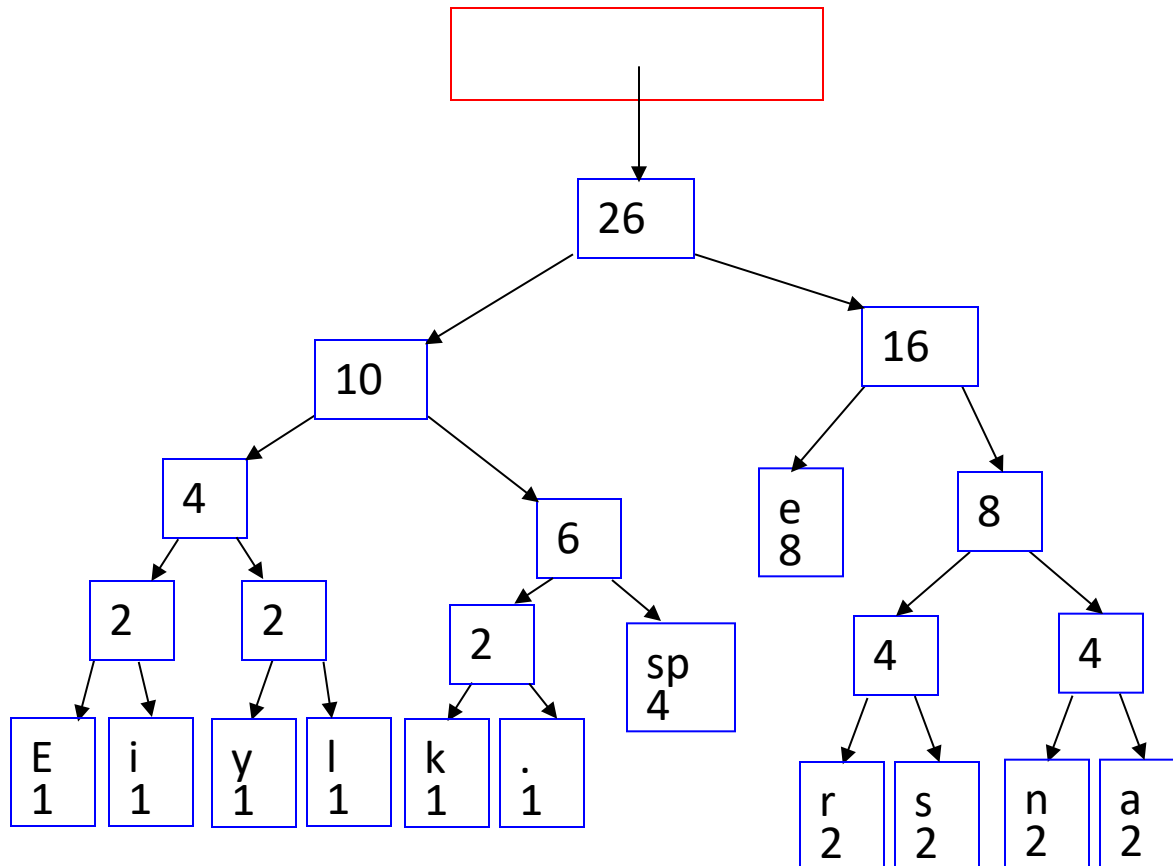


# Cont...





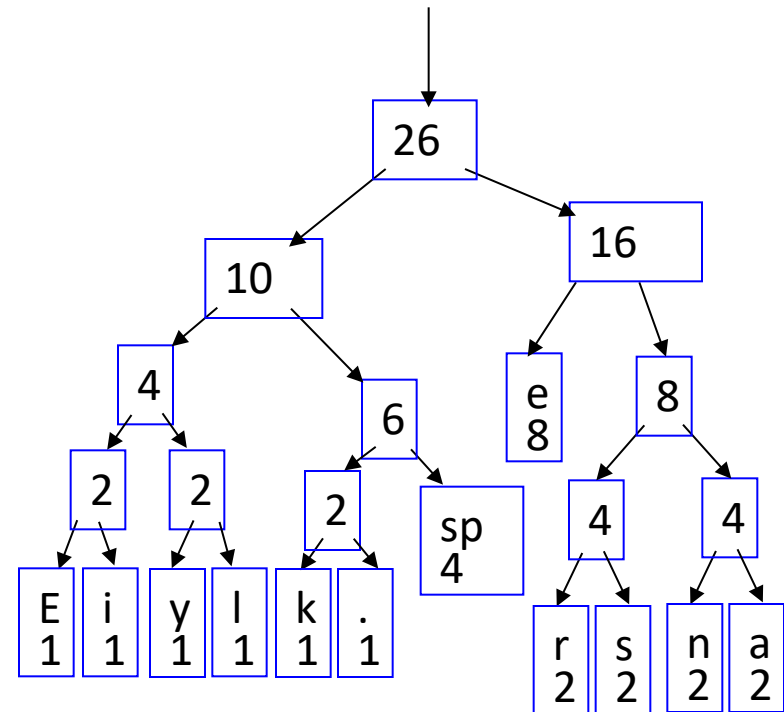
# Cont...



# Cont...

- Going left is a 0 going right is a 1

Char	Code
E	0000
i	0001
y	0010
l	0011
k	0100
.	0101
space	011
e	10
r	1100
s	1101
n	1110
a	1111



# Thanks!

Figure and slide materials are taken from the following sources:

1. W. Stallings, (2010), [Data and Computer Communications](#)
2. [NPTL lecture](#) on Data Communication, by Prof. A. K. Pal, IIT Kharagpur
3. B. A. Forouzan, (2013), [Data Communication and Networking](#)