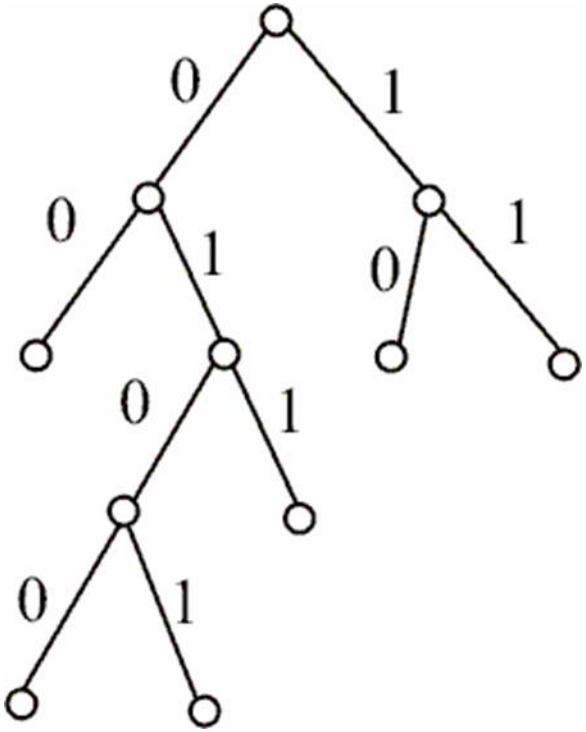
PREFIX CODES

- Bit string is often used in computer and communication areas.
- if e were encoded with 0, a with 1, and t with 01, then the bit string 0101 could correspond to eat, tea, eaea, or tt.
- **prefix codes:** the bit string for a letter never occurs as the first part of the bit string for another letter.
- {1,00,011,0101,01001,01000}
- {1,00,011,0101,0100,01001,01000}

- A prefix code can be represented using a binary tree, where the characters are the labels of the leaves in the tree.
- The edges of the tree are labeled so that an edge leading to a left child is assigned a 0 and an edge leading to a right child is assigned a 1.
- The bit string used to encode a character is the sequence of labels of the edges in the unique path from the root to the leaf that has this character as its label.

example

- { 00, 10, 11, 011, 0100, 0101 }



Huffman algorithm

1. begins with a forest of trees each consisting of one vertex, where each vertex has a symbol as its label and where the weight of this vertex equals the frequency of the symbol that is its label.

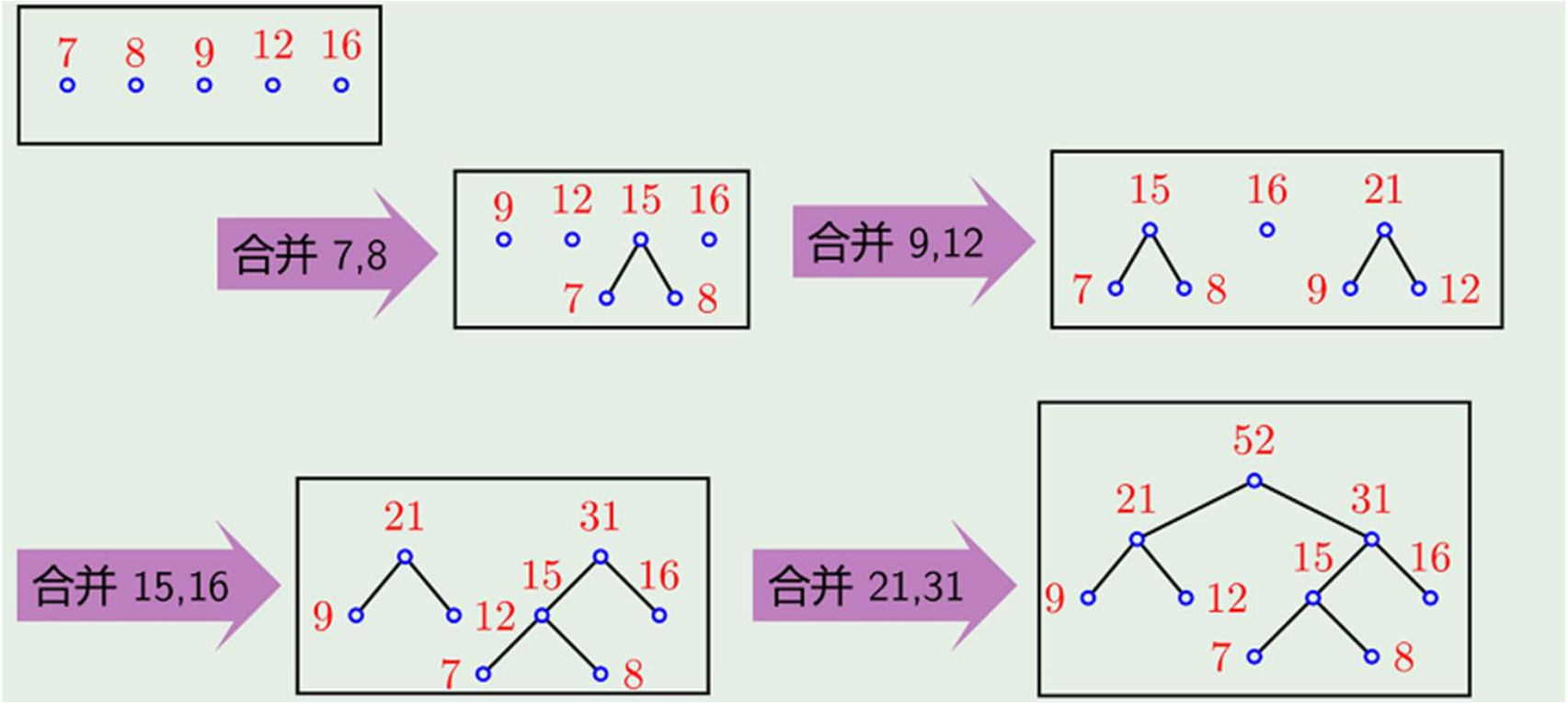
2. At each step, we combine two trees having the least total weight into a single tree by introducing a new root and placing the tree with smaller weight as its left subtree and the tree with larger weight as its right subtree.

3. Furthermore, we assign the sum of the weights of the two subtrees of this tree as the total weight of the tree.

4. The algorithm is finished when the forest is reduced to a single tree.

example

Construct a binary tree where the weights of leaves are 7,8,9,12,16 using Huffman algorithm



Huffman coding

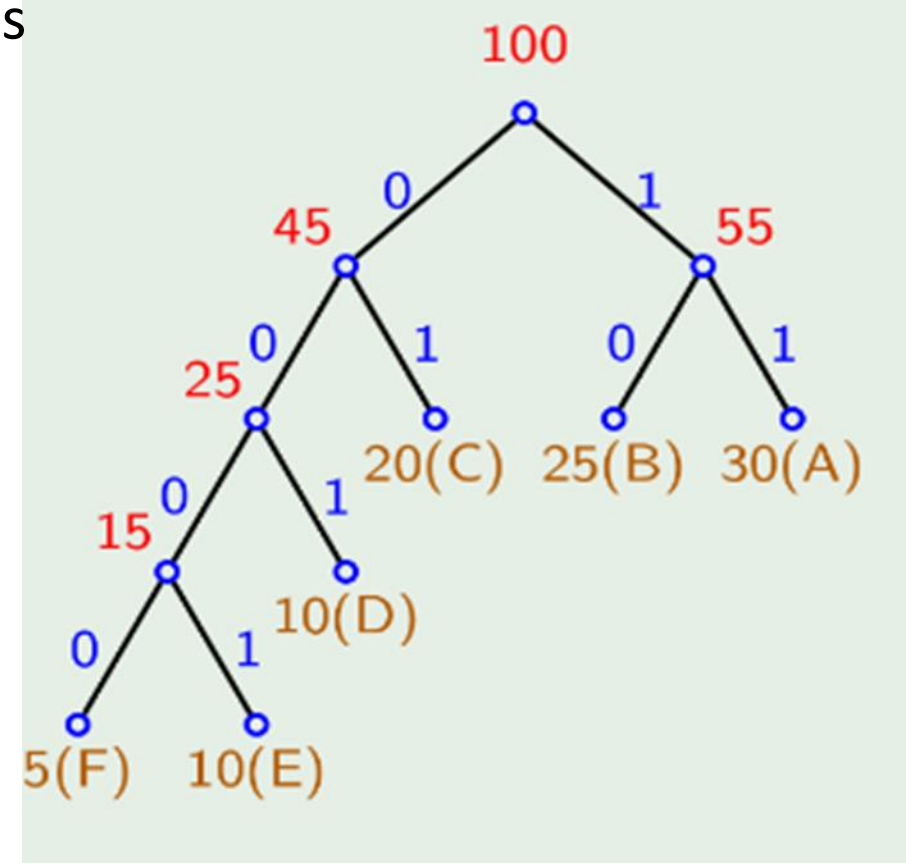
Use Huffman coding to encode the following symbols with the frequencies listed：

A：30%　　B：25%

C：20%　　D：10%

E：10%　　F：5%

What is the average number of bits used to encode a character?



Tree Traversal

- Ordered rooted trees are often used to store information.
- We usually visit each vertex in order
- Check if some vertices has properties we are interested in, or get some information for some vertices to do some operations, etc.
- Tree traversal is the procedure for visiting each vertex of an ordered rooted tree .

TRAVERSAL ALGORITHMS

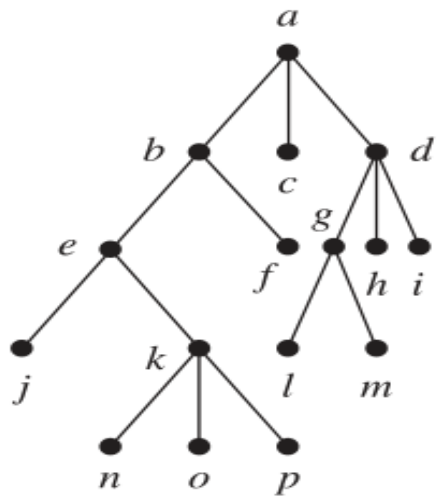
Procedure for systematically visiting each vertex of an ordered rooted tree are called traversal algorithms.

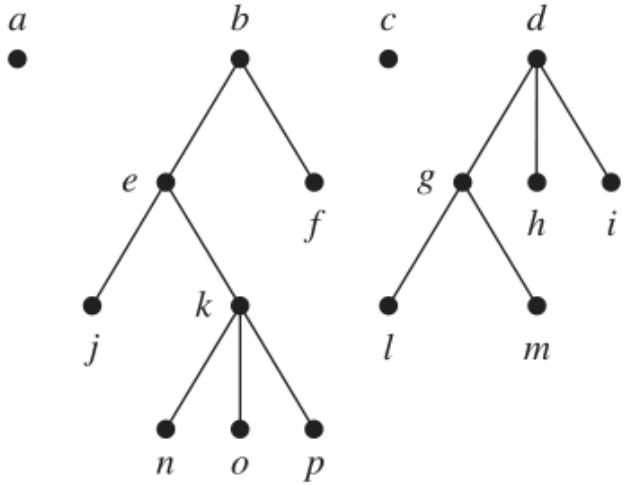
- preorder traversal
- inorder traversal
- postorder traversal

Preorder traversal

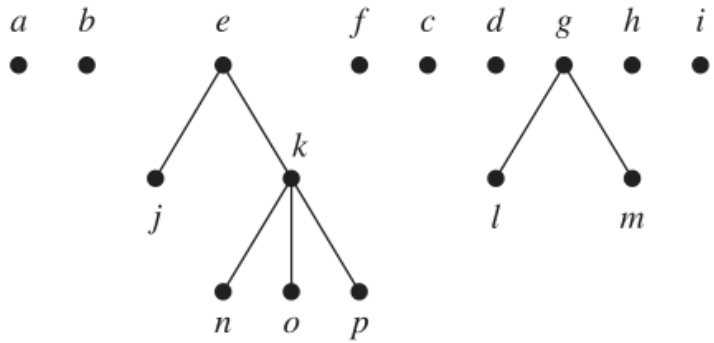
- **Definition 1:** Let T be an ordered rooted tree with root r. If T consists only of r, then r is the preorder traversal of T. Otherwise, suppose that $T_1, T_2, \ldots, T_n$ are the subtrees at r from left to right in T. The preorder traversal begins by visiting r. It continues by traversing T_1 in preorder, then T_2 in preorder, and so on, until T_n is traversal in preorder.

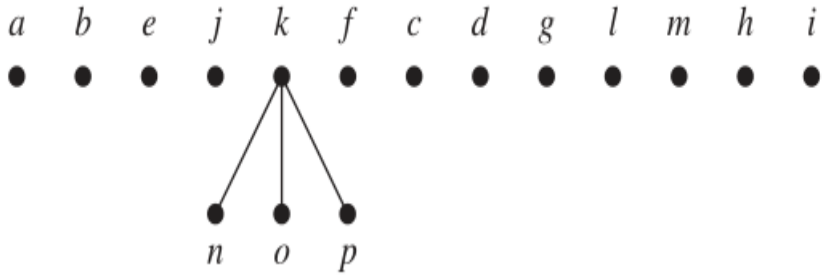
Example

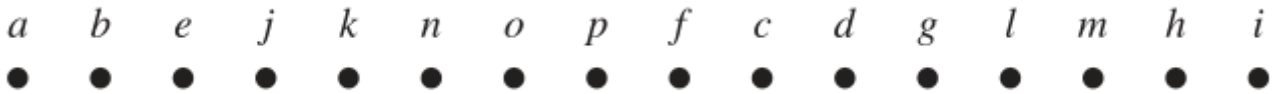

a
b
c
d
e
f
g
h
i
j
k
l
m
n
o
p


a
b
e
f
j
k
n
o
p

c
d
g
h
i
l
m


a
b
e
f
c
d
g
h
i
j
k
l
m
n
o
p

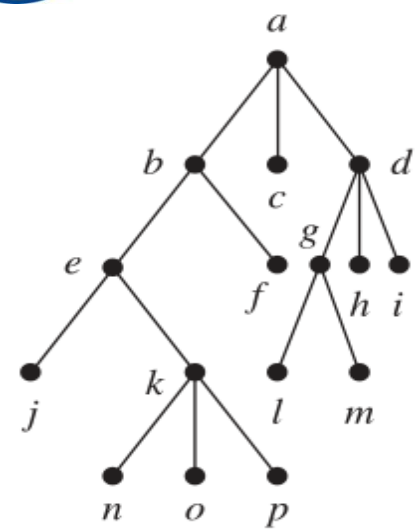

a
b
e
j
k
f
c
d
g
l
m
h
i
n
o
p

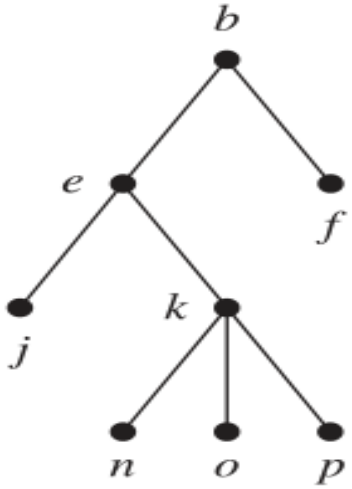

a
b
e
j
k
n
o
p
f
c
d
g
l
m
h
i

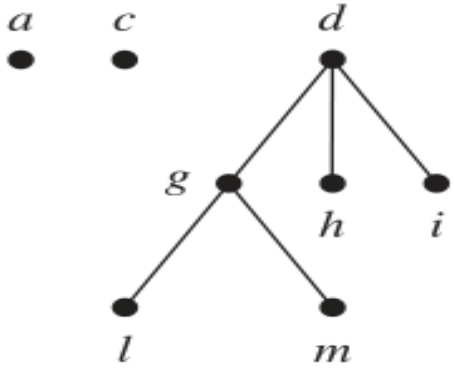
inorder traversal

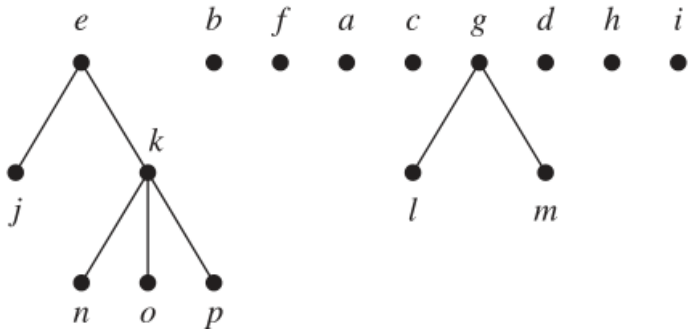
Definition 2: Let T be an ordered rooted tree with root r. If T consists only of r, then r is the *inorder traversal* of T. Otherwise, suppose that $T_1, T_2, \ldots, T_n$ are the subtrees at r from left to right in T.The *inorder traversal* begins by traversing T_1 in inorder then visiting r. It continues by traversing T_2 in inorder, then and T_3 in inorder , and finally T_n in inorder.

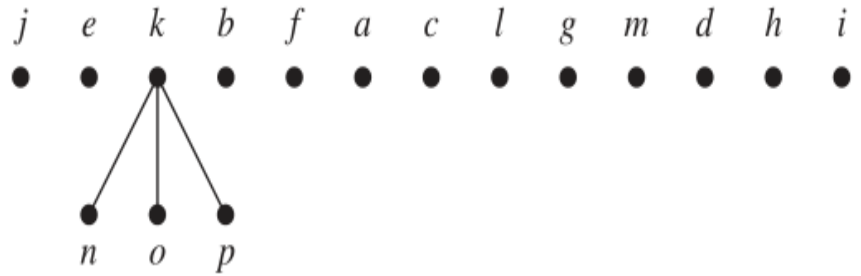
Example

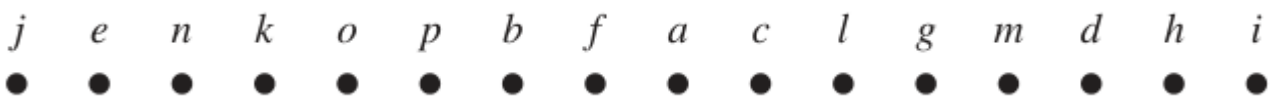

a
b
c
d
e
f
g
h
i
j
k
l
m
n
o
p


b
e
f
k
j
n
o
p


a
c
d
g
h
i
l
m


e
b
f
a
c
g
d
h
i
k
j
l
m
n
o
p

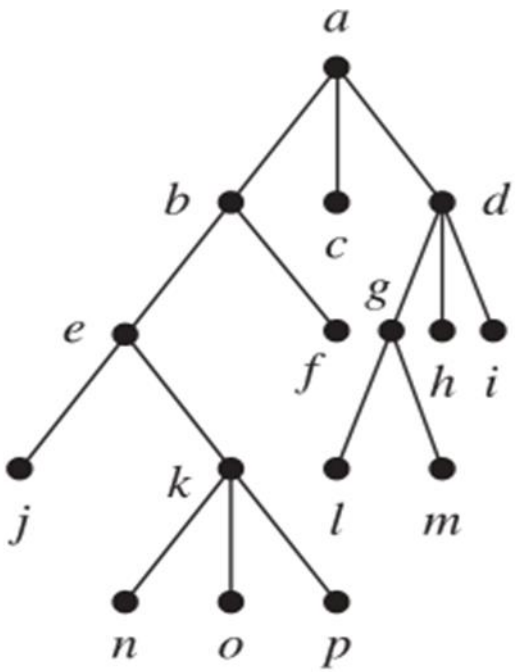

j
e
k
b
f
a
c
l
g
m
d
h
i
n
o
p


j
e
n
k
o
p
b
f
a
c
l
g
m
d
h
i

postorder traversal

- **Definition 3:** Let T be an ordered rooted tree with root r. If T consists only of r, then r is the *postorder traversal* of T. Otherwise, suppose that $T_1, T_2, \ldots, T_n$ are the subtrees at r from left to right in T.The *postorder traversal* begins by traversing T_1 in postorder, then T_2 in postorder, then T_n in postorder, and ends by visiting r.

- The preorder traversal of T is *j,n,o,p,k,e,f,b,c,l,m,g, h,i,d,a*



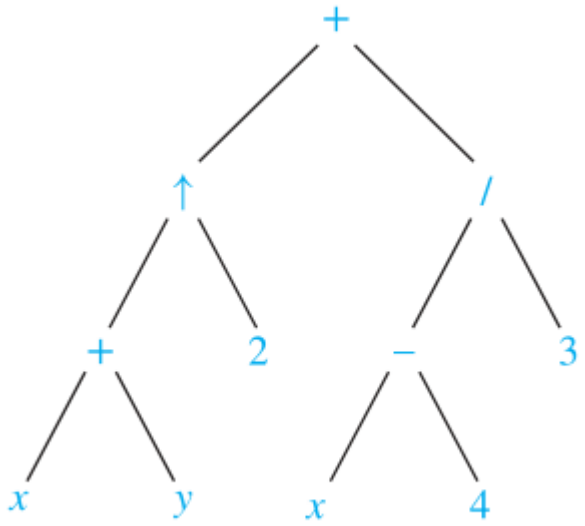
Infix, prefix and postfix notation

- An ordered rooted tree can be used to represent complicated expressions, such as compound propositions, combinations of sets, and arithmetic expressions , where the internal vertices represent operations, and the leaves represent the variables or numbers.

Example

$((x + y)\uparrow 2) + ((x - 4)/3)$

- If we visit every vertex in inorder, we can get this expression.
- because of precedence of these operations, parentheses must be used to specify order.

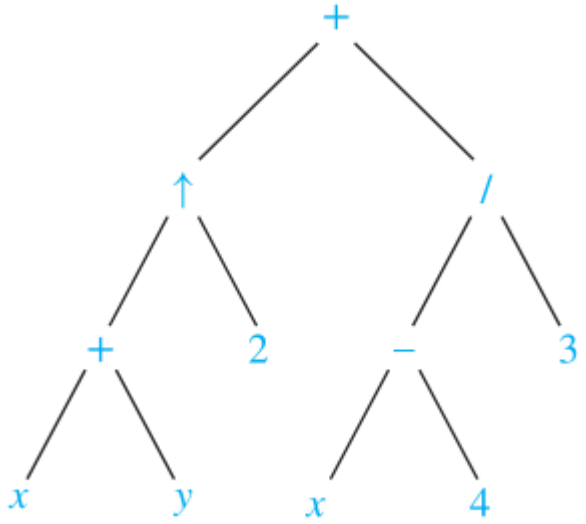


prefix notation

- $((x + y)\uparrow 2) + ((x - 4)/3)$

The prefix form of this expression is

$+\uparrow+xy2/-\ x43$



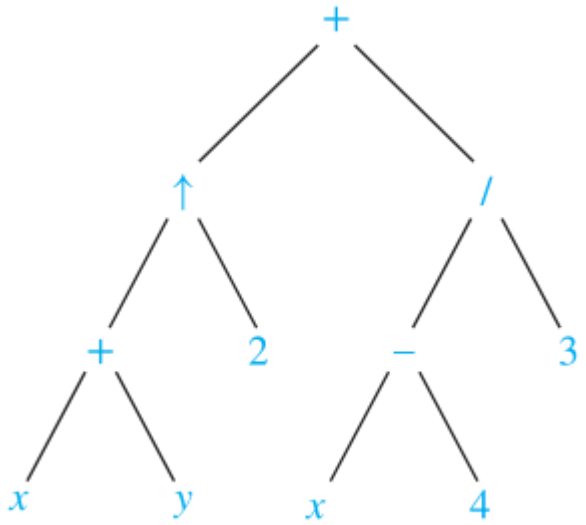
- In the prefix form of an expression, a binary operator precedes its two operands.
- we get an expression in prefix form from right to left.
- perform the corresponding operation with the two operands to the right of this operand

postfix notation

- $((x + y)\uparrow 2) + ((x - 4)/3)$

The postfix form of this expression is

$xy+ 2\uparrow x4 - 3 / +$



- we get an expression in prefix form from left to right.
- perform the corresponding operation with the two operands to the left of this operand

homework

- 11.2 P806

20,21,22,23,24

- 11.3 P820

8,14,18,23(b),24(b)