SC1007
Data Structures and Algorithms

Huffman Coding Tree

Dr. Loke Yuan Ren

Lecturer

yrloke@ntu.edu.sg

College of Engineering

School of Computer Science and Engineering

Huffman Coding

- A greedy algorithm
    - The coding tree builds up based on the frequency of occurrence of each character
- Variable-length Code
    - More frequently used characters are represented by fewer bits
    - ASCII is fixed-length code (7-bit)
- Prefix Code
    - No codeword is also a prefix of some other codeword (binary string)
        - {0,10,110,111} is a prefix code
        - {0,01,110,111} => 0 and 01 are not prefix-free codes
- Lossless data compression

- Developed by David A. Huffman in 1952

Why do we need Huffman Coding?

	a	b	c	d	e	f
Frequency	0.45	0.13	0.12	0.16	0.09	0.05
Fixed Length codeword	000	001	010	011	100	101
Variable-length codeword	0	101	100	111	1101	1100

If you have 100k characters to store,

300k bits are required for fixed-length codeword

224k bits are required for variable-length codeword

Saving of ~25% space

0.45*1+0.13*3+0.12*3+0.16*3+0.09*4+0.05*4 = 2.24

Why do we need Huffman Coding?

	a	b	c	d	e	f
Frequency	0.45	0.13	0.12	0.16	0.09	0.05
Variable-length codeword	0	101	100	111	1101	1100

Prefix Code aka prefix-free codes or comma-free code

011111011011101

Why do we need Huffman Coding?

	a	b	c	d	e	f
Frequency	0.45	0.13	0.12	0.16	0.09	0.05
Variable-length codeword	0	101	100	111	1101	1100

0111110110111101

a d e b e

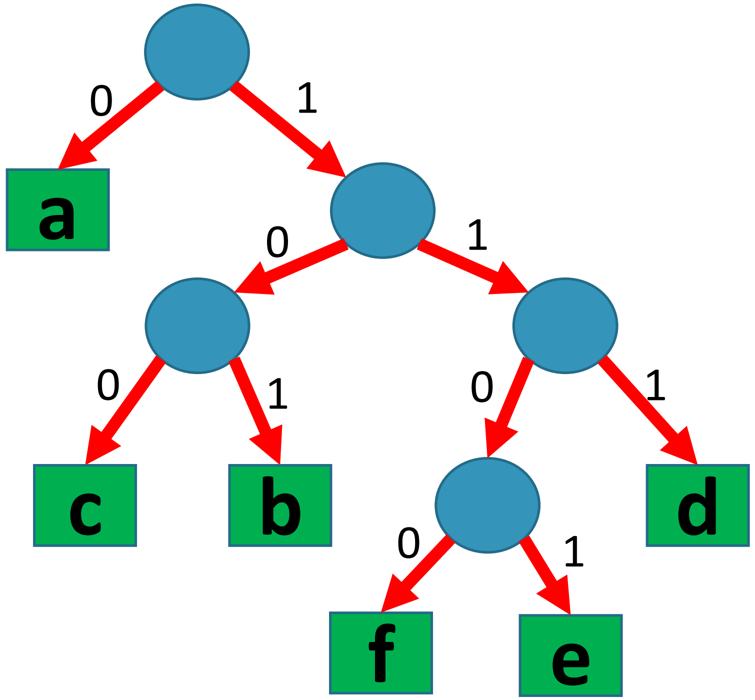
You are not able to have an ambiguous code from Huffman coding.

How does Huffman codes work?

	a	b	c	d	e	f
Frequency	0.45	0.13	0.12	0.16	0.09	0.05
Variable-length codeword	0	101	100	111	1101	1100

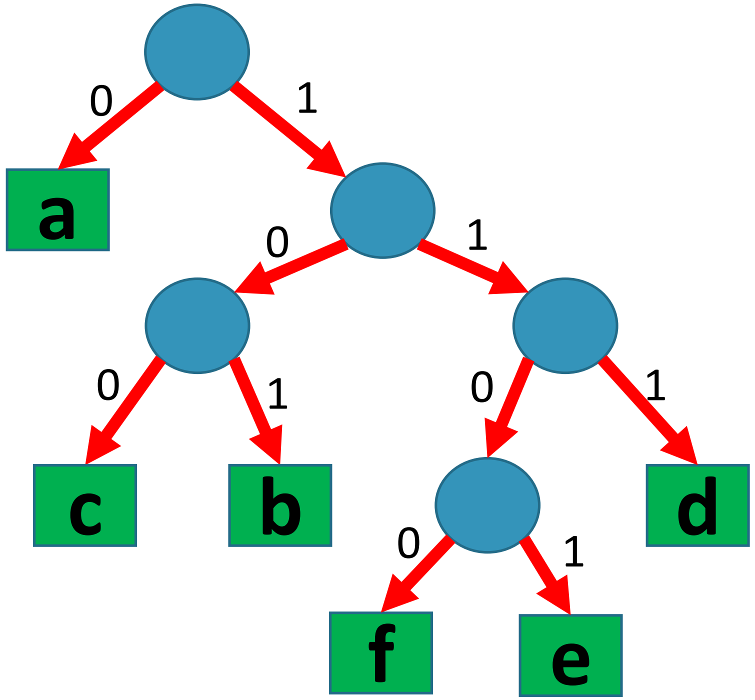
0111110110111101

a d e b e



How does Huffman codes work?

	a	b	c	d	e	f
Frequency	0.45	0.13	0.12	0.16	0.09	0.05
Variable-length codeword	0	101	100	111	1101	1100

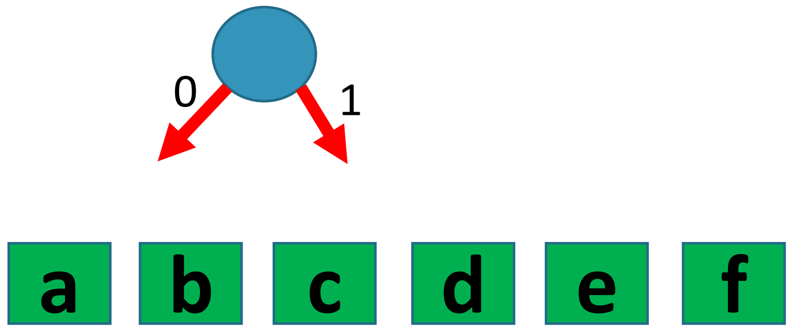


How does Huffman codes work?

	a	b	c	d	e	f
Frequency	0.45	0.13	0.12	0.16	0.09	0.05
Variable-length codeword	0	101	100	111	1101	1100

Huffman Coding Tree

```
Create nodes to all data (characters) with their frequency
for i ← 1 to n-1
    Create a node z
    Find a node x with the smallest frequency f(x)
    Find a node y with the second smallest frequency f(y)
    Add their frequency f(z) = f(x) + f(y)
    Construct a subtree which the root is z, left child is x and right child is y
    Add z into the node list
```

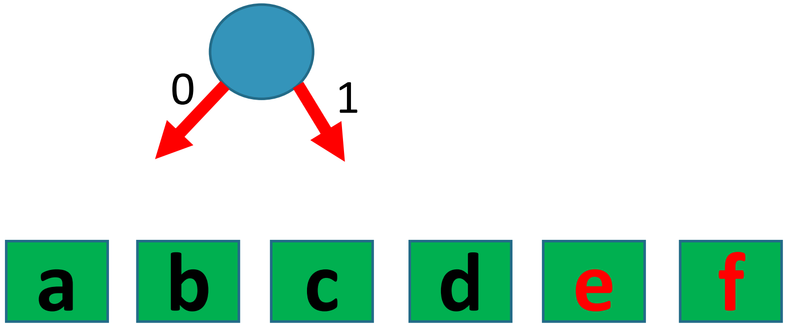


How does Huffman codes work?

	a	b	c	d	e	f
Frequency	0.45	0.13	0.12	0.16	0.09	0.05
Variable-length codeword	0	101	100	111	1101	1100

Huffman Coding Tree

```
Create nodes to all data (characters) with their frequency
for i ← 1 to n-1
    Create a node z
    Find a node x with the smallest frequency f(x)
    Find a node y with the second smallest frequency f(y)
    Add their frequency f(z) = f(x) + f(y)
    Construct a subtree which the root is z, left child is x and right child is y
    Add z into the node list
```

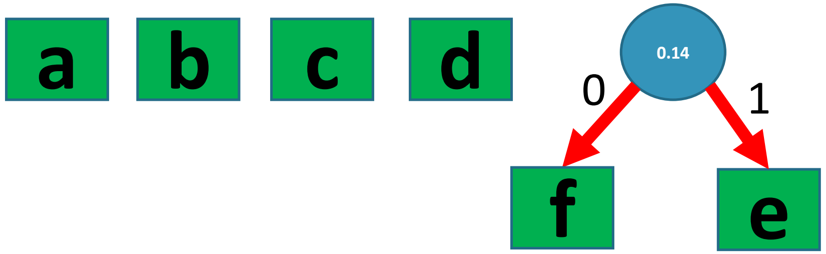


How does Huffman codes work?

	a	b	c	d	e	f
Frequency	0.45	0.13	0.12	0.16	0.09	0.05
					0.14	
Variable-length codeword	0	101	100	111	1101	1100

Huffman Coding Tree

```
1. Create nodes to all data (characters) with their frequency
2. for i ← 1 to n-1
3.         Create a node z
4.         Find a node $x$ with the smallest frequency $f(x)$
5.         Find a node $y$ with the second smallest frequency $f(y)$
6.         Add their frequency $f(z) = f(x) + f(y)$
7.         Construct a subtree which the root is z, left child is x and right child is y
8.         Add z into the node list
```

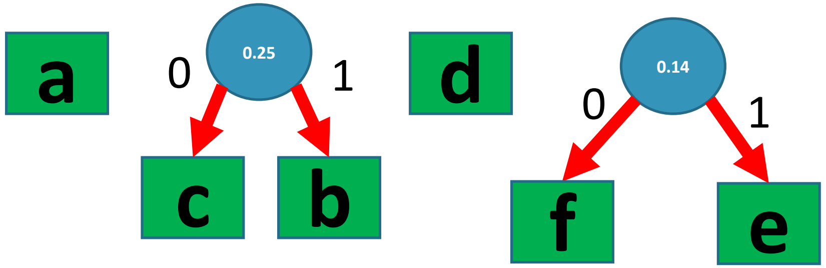


How does Huffman codes work?

	a	b	c	d	e	f
Frequency	0.45	0.13	0.12	0.16		
		0.25			0.14	
Variable-length codeword	0	101	100	111	1101	1100

Huffman Coding Tree

```
Create nodes to all data (characters) with their frequency
for i ← 1 to n-1
    Create a node z
    Find a node *x* with the smallest frequency *f(x)*
    Find a node *y* with the second smallest frequency *f(y)*
    Add their frequency *f(z) = f(x) + f(y)*
    Construct a subtree which the root is z, left child is x and right child is y
    Add z into the node list
```

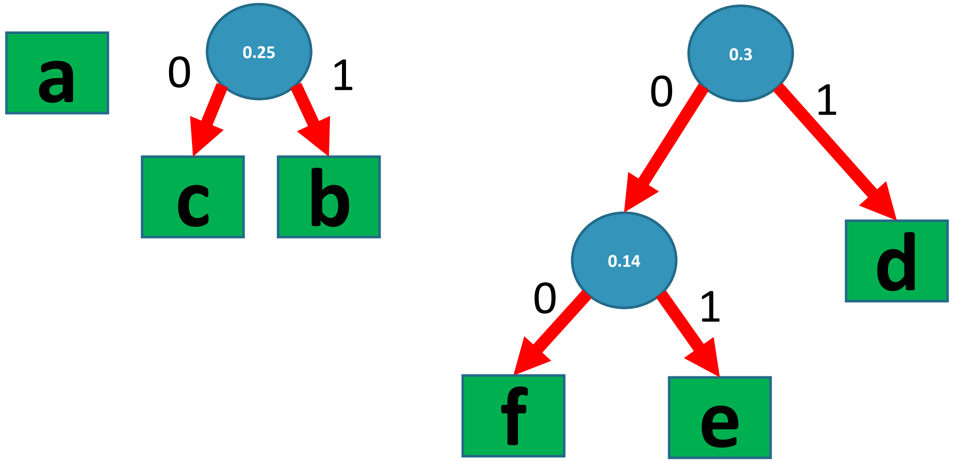


How does Huffman codes work?

	a	b	c	d	e	f
Frequency	0.45	0.25		0.16	0.14	
				0.3		
Variable-length codeword	0	101	100	111	1101	1100

Huffman Coding Tree

```
Create nodes to all data (characters) with their frequency
for i ← 1 to n-1
    Create a node z
    Find a node x with the smallest frequency $f(x)$
    Find a node y with the second smallest frequency $f(y)$
    Add their frequency $f(z) = f(x) + f(y)$
    Construct a subtree which the root is z, left child is x and right child is y
    Add z into the node list
```

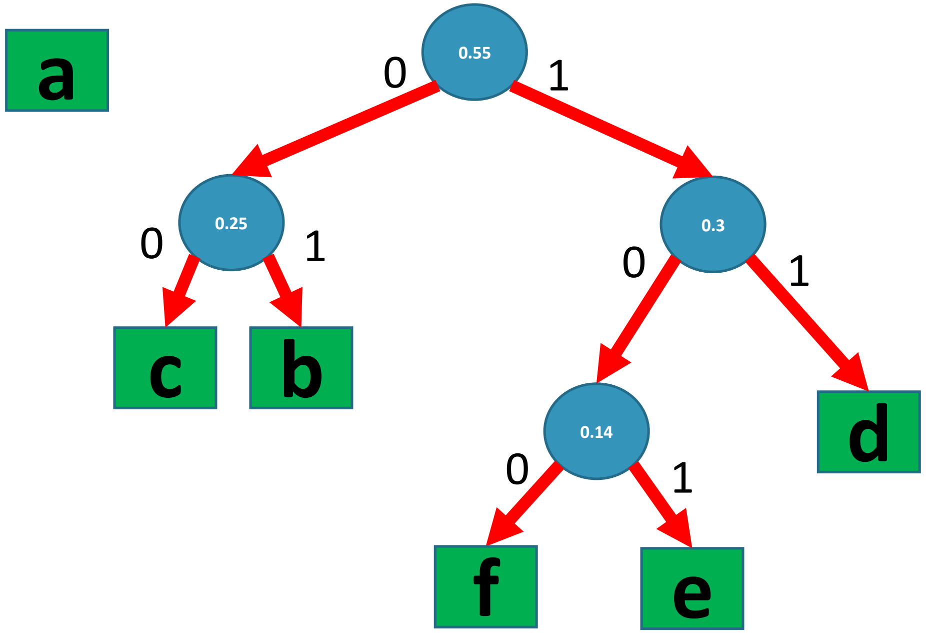


How does Huffman codes work?

	a	b	c	d	e	f
Frequency	0.45	0.25		0.3		
		0.55				
Variable-length codeword	0	101	100	111	1101	1100

Huffman Coding Tree

```
Create nodes to all data (characters) with their frequency
for i ← 1 to n-1
    Create a node z
    Find a node x with the smallest frequency f(x)
    Find a node y with the second smallest frequency f(y)
    Add their frequency f(z) = f(x) + f(y)
    Construct a subtree which the root is z, left child is x and
  right child is y
    Add z into the node list
```



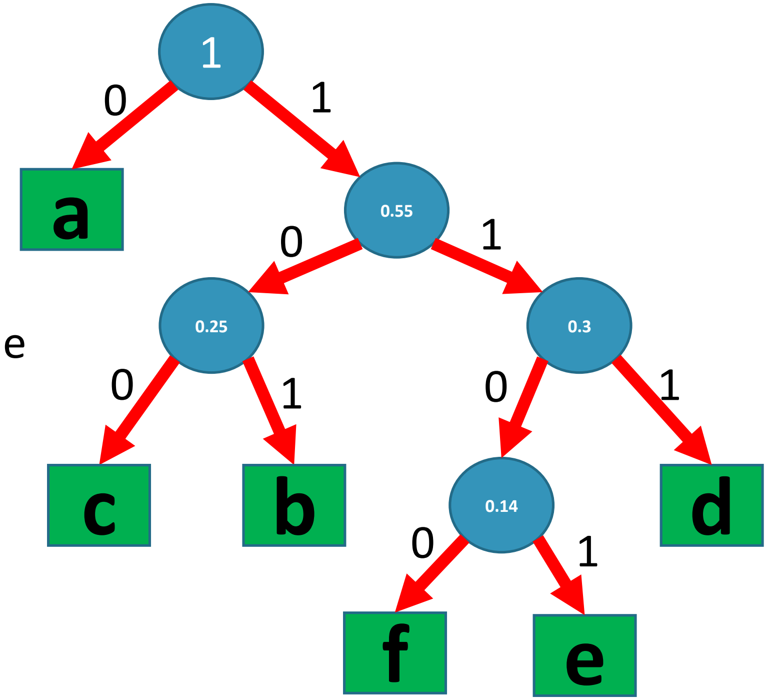
How does Huffman codes work?

	a	b	c	d	e	f
Frequency	0.45	0.13	0.12	0.16	0.09	0.05
Variable-length codeword	0	101	100	111	1101	1100

0111110110111101

a d e b e

A Radix Tree: The bit strings indicate the traversal path from root to the node


1
0
1
a
0.55
0
1
0.25
0.3
0
1
0
1
c
b
0.14
d
0
1
f
e