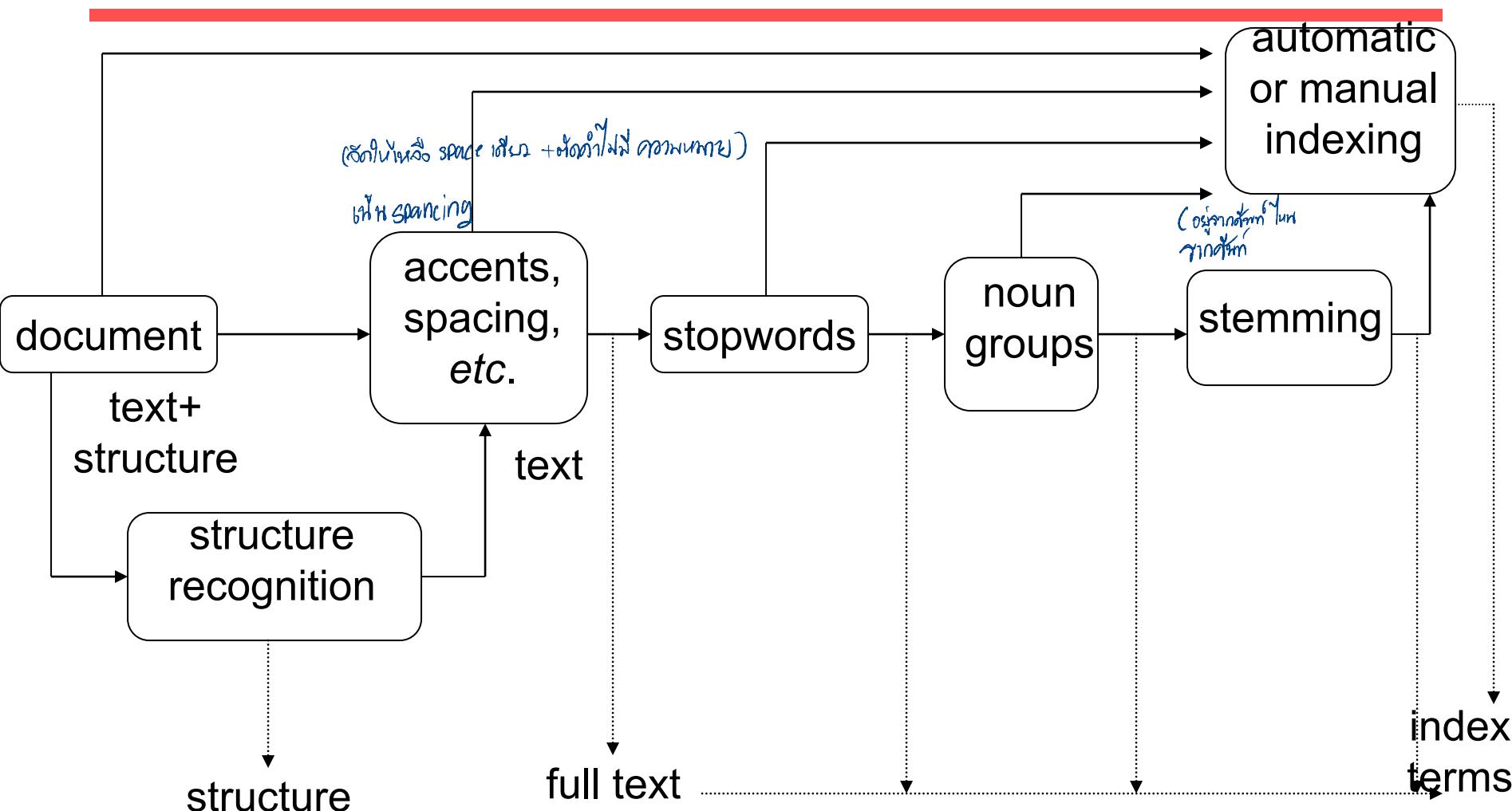

Chapter 6

Text Operations

Logical View of a Document



Text Operations

- Lexical analysis of the text รีเกกชันฟอร์ม
- Elimination of stopwords ตัวอักขระ...
- Stemming ตัดเส้น
- Selection of index terms หัวเรื่อง หัวข้อ
- Construction of term categorization structures โครงสร้างของคำ

Lexical Analysis of the Text

- Word separators

ໃໝ່ eng

- space ໄກນທີ່ຈະໄສ space ຕາມການ

- digits ກາບຕົວ ໂພນ ເຊັ່ນ (ໄຟລືດອຍເຄີຍ keyword) ໃຫ້ເວົ້າການ ອົບ

- hyphens - (ເຊື່ອງຫຼືກ່າ - ນາຄົ່າ ລົ້ອດອາວຸນ)

- punctuation marks ເກົ່ານໍ້າມຍາກພາບພາວ

- the case of the letters ດີວ່າລົກ / ດີວ່າບູນ ; ຖໍ່ໄປຢ່າງເປົ້າເປົ້າກຳນົດ

Lexical Analysis for Automatic Indexing

- **Lexical Analysis**

onclusion of token

*Convert an input stream of characters into stream **words** or **token**.*

- **What is a word or a token?**

Tokens consist of letters.

- digits: Most numbers are not good index terms.

counterexamples: case numbers in a legal database, “B6” and “B12” in vitamin database.

Lexical Analysis for Automatic Indexing

(Continued)

- hyphens – นำสิ่งที่มีเส้นหักมาแยก

 - break hyphenated words: state-of-the-art, state of the art
 - keep hyphenated words as a token: “Jean-Claude”, “F-16”

- punctuation marks: often used as parts of terms, e.g., OS/2, 510B.C.
- case: usually not significant in index terms

例句: สีสันเดื่อยักษ์ เตี้ยๆเดื่อยักษ์ → เดื่อยักษ์ + เดื่อยักษ์

(อาจมีข้อความที่รวม ให้ความรู้สึกตื่นเต้น)
ตัวลักษณะ ที่มีค่าน้ำหนักมาก

Elimination of Stopwords

- A list of stopwords
 - words that are too frequent among the documents
 - articles, prepositions, conjunctions, etc.
- Can reduce the size of the indexing structure considerably
- Problem
 - Search for “**to be or not to be**”?

Stopword

- Avoid retrieving almost every item in a database regardless of its relevance.
- Examples eng 425 ជាកំណត់ស្ថាបន្ទាន់ និងរបៀប
 - conservative approach (ORBIT Search Service): *and, an, by, from, of, the, with*
 - (derived from Brown corpus): 425 words *a, about, above, across, after, again, against, all, almost, alone, along, already, also, although, always, among, an, and, another, any, anybody, anyone, anything, anywhere, are, area, areas, around, as, ask, asked, asking, asks, at, away, b, back, backed, backing, backs, be, because, became, ...*
- Articles, prepositions, conjunctions, ...

Stemming

ស្ថិកសម្រាប់
ការបញ្ចូលការពិនិត្យ
សម្រាប់ការសរសៃរដ្ឋភាសា

- Example
 - *connect, connected, connecting, connection, connections*
 - effectiveness --> effective --> effect
 - picnicking --> picnic
- Removing strategies
 - affix removal: intuitive, simple
 - table lookup តាម (ការសរសៃ)
 - successor variety រដ្ឋភាសា (ការពិនិត្យ)
 - n-gram (ជាពាណ)

Stemmers

- programs that relate morphologically similar indexing and search terms
 - stem at indexing time
 - advantage: efficiency and index file compression
 - disadvantage: information about the full terms is lost
 - example (CATALOG system), *stem at search time*
 - Look for: system users
 - Search Term: users
- | Term | Occurrences |
|----------|-------------|
| 1. user | 15 |
| 2. users | 1 |
| 3. used | 3 |
| 4. using | 2 |

The user selects the terms he wants by numbers

Conflation Methods

- manual
- automatic (stemmers)
 - affix removal
 - longest match vs. simple removal
 - successor variety
 - table lookup
 - n-gram
- evaluation
 - correctness
 - retrieval effectiveness
 - compression performance



Term	Stem
engineering	engineer
engineered	engineer
engineer	engineer

ຄົນນັ້ນ

ຈິງຈາກ level

Successor Variety

- **Definition**

ອຳນວຍດີນີ້ໃນ Doc / ຕະຫຼາມ

the number of different characters that follow it in words in some body of text

- **Example**

a body of text: *able, axle, accident, ape, about*

successor variety of *apple*

1st: 4 (*b, x, c, p*)

2nd: 1 (*e*)

Successor Variety (Continued)

កិច្ចសម្រាប់ ទូរសព្ទអង្គភាព

- **Idea**

The successor variety of substrings of a term will decrease as more characters are added until a segment boundary is reached, i.e., the successor variety will **sharply increase**.

- **Example**

Test word: READABLE

លទ្ធផលរបស់ការសរសៃរាយ A MJ Z នៅក្នុងអង្គភាព

Corpus: ABLE, BEATABLE, FIXABLE, READS,
READABLE, READING, RED, ROPE, RIPE

Prefix	Successor Variety	Letters
R	3 (RE, RO, RI)	E, O, I
RE	2 (REA, RED)	A, D
REA	1 (READ)	D
READ	3 (...S, ...A, ...I)	A, I, S
READA	1	B
READAB	1	L
READABL	1	E
READABLE	1	blank

↓ օրուան 2-gram, 3-gram օ. օրէ 2-gram

n-gram stemmers

ກໍາ 2 ດຳ ໄກສ໌ ເສີມກຳນົດຍຸ ໄຕ້ ປັນ

- **diagram**
a pair of consecutive letters
- **shared diagram method**
association measures are calculated between pairs of terms

$$S = \frac{2C}{A + B}$$

where A: *the number of unique diagrams in the first word,*
B: *the number of unique diagrams in the second,*
C: *the number of unique diagrams shared by A and B.*

n-gram stemmers (Continued)

• Example

2-gram ① หนึ่งมาก็: 2 ตัวอักษร ↓

คำที่ 1

statistics => st ta at ti is st ti ic cs

1 ตัวอักษร กับ 2 ตัวอักษร +

ผู้คนต้องการที่จะเขียน
A-Z

คำที่ 2

unique diagrams => at cs ic is st ta ti

statistical => st ta at ti is st ti ic ca al

unique diagrams => al at ca ic is st ta ti

(ตัวอักษร กับ 2 ตัวอักษร)

$$S = \frac{2C}{A + B} = \frac{2 * 6}{7 + 8} = 0.80$$

ผลลัพธ์ที่ได้ (ใจ). ใช่ เช่น)

ตัวเลขที่ได้ 0.80 เท่ากับ 80% = ประมาณ 80%

ตัวอักษรที่ได้

n-gram stemmers (Continued)

- **similarity matrix** *9 នៃ n matrix*

determine the semantic measures for all pairs of terms in the database

	word ₁	word ₂	word ₃	...	word _{n-1}
word ₁					
word ₂		S ₂₁			
word ₃	S ₃₁		S ₃₂		
.					
.					
word _n	S _{n1}	S _{n2}	S _{n3}	...	S _{n(n-1)}

- terms are clustered using a single link clustering method
- more a term clustering procedure than a stemming one

Affix Removal Stemmers

- **procedure**

Remove suffixes and/or prefixes from terms leaving a stem, and transform the resultant stem.

- **example:** plural forms *en eng a:ldmannarðir*
If a word ends in “ies” but not “eies” or “aies”
 then “ies” --> “y”
If a word ends in “es” but not “aes”, “ees”, or
“oes”
 then “es” --> “e”
If a word ends in “s”, but not “us” or “ss”
 then “s” --> NULL

Index Terms Selection

ເລືອກເນື້ອທຸລະນ

- Motivation
 - A sentence is usually composed of nouns, pronouns, articles, verbs, adjectives, adverbs, and connectives.
 - Most of the **semantics** is carried by the **noun** words.
- Identification of noun groups
 - A noun group is a set of nouns whose syntactic distance in the text does not exceed a predefined threshold

Index Terms Selection

- Indexing by single words
 - single words are often ambiguous and not specific enough for accurate discrimination of documents
 - **bank terminology vs. terminology bank**
- Indexing by phrases
 - Syntactic phrases are almost always more specific than single words
- Indexing by single words and phrases

ເຊື້ອທີ່ຕັ້ງກັນພາກກໍາລະວົງ (ເປົ້າຫຼັກ)
 keyword
ພາກກໍາລະວົງ
ພາກກໍາລະວົງ

(ຄວາມສຳເນົາຂອງພາກ)

Thesauri

- Peter Roget, 1988
- Example

cowardly adj. คื่นชื่น

คำเดียวกัน อย่างไร Dic นี่ = ตัวอักษร เช่น]

Ignobly lacking in courage: cowardly
turncoats

คำนี้มีความหมายเดียวกัน Dic = ความกลัว เดือดร้อนมาก

Syns: chicken (slang), chicken-hearted,
craven, dastardly, faint-hearted, gutless,
lily-livered, pusillanimous, unmanly, yellow
(slang), yellow-bellied (slang).

- A controlled vocabulary for the indexing and searching

The Purpose of a Thesaurus

ជំនួយអាមេរិក និងអាស៊ាន ដើម្បីបង្កើតសារព័ត៌មាន

- To provide a **standard** vocabulary for indexing and searching
- To assist users with locating terms for proper query formulation
- To provide classified hierarchies that allow the broadening and narrowing of the current query request

Functions of thesauri

- Provide a standard vocabulary for indexing and searching
- Assist users with locating terms for proper query formulation
- Provide classified hierarchies that allow the broadening and narrowing of the current query request

Usage

- **Indexing**
 - Select the most appropriate thesaurus entries for representing the document.
- **Searching**
 - Design the most appropriate search strategy.
 - If the search does **not retrieve enough** documents, the thesaurus can be used to **expand the query**.
ស្ថាប័នកុំព្យូទ័រ
 - If the search retrieves **too many** items, the thesaurus can suggest **more specific search vocabulary**.
ស្ថាប័នកុំព្យូទ័រដើម្បីបង្កើតការស្លាកស្ថាប័ន

Document Clustering

Document Clustering

- Global clustering
 - The grouping of documents accordingly to their occurrence in the whole collection
- Local clustering:
 - The grouping of the local set of retrieved documents by a query

Text Compression

- Finding ways to represent the text in fewer bits or bytes จำนวน byte น้อยลง
- Encode/Decode
- Low data communication

Text Compression

- Lossless Compression

- Text 



- Lossy Compression 

- Multimedia 

AI detect ความถูกต้อง = ความเร็ว

All case **Speed** is important !!!

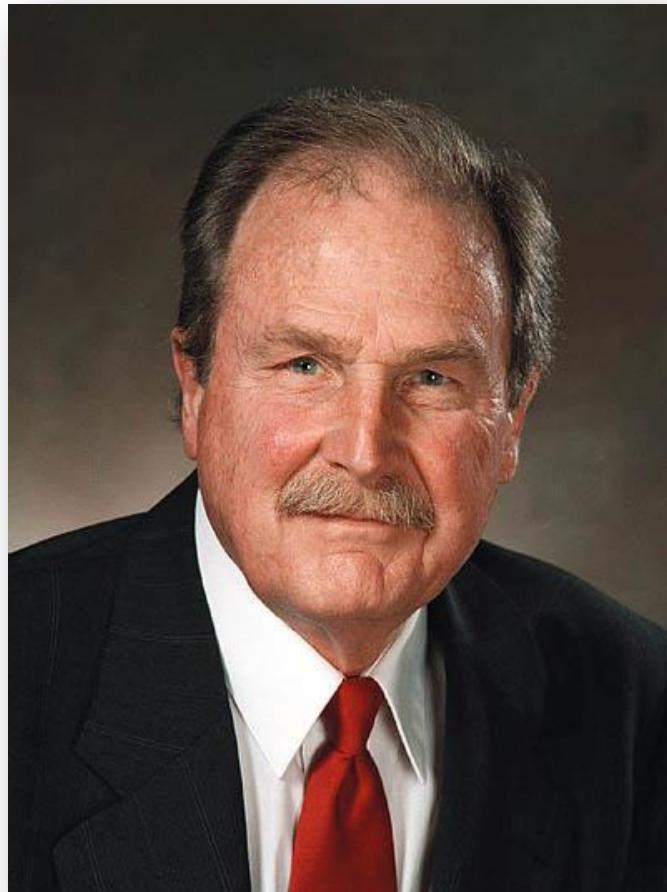
Speed depend on

- Frequency Compute
- Tree Creation 
- Dictionary Creation/Using

Text Compression

- Statistical Methods
 - Huffman coding
- Dictionary-based
 - Ziv-Lempel

Huffman Coding



Huffman Coding

จัดทำโดย David A. Huffman

เป็นวิธีการเข้ารหัส Entropy ชนิดหนึ่ง

ที่ใช้ในการบีบอัดข้อมูลแบบ

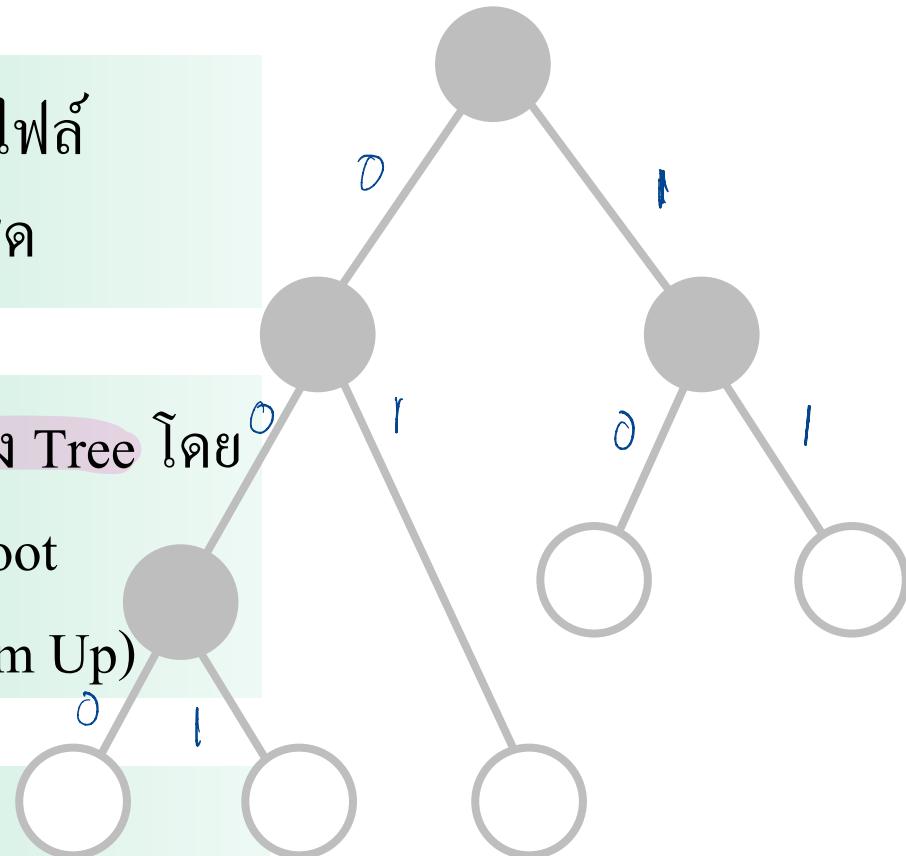
lossless data compression

Huffman Coding

Huffman Coding เป็นวิธีการบีบอัดไฟล์ที่ให้ผลลัพธ์มีขนาดเล็กลงมากที่สุด

วิธีการของ Huffman นี้ ทำการสร้าง Tree โดยเริ่มจาก Leaf Node ไปหา Root หรือสร้างจากล่างขึ้นบน (Bottom Up)

รหัสที่สร้างโดยวิธีของ Huffman จะเป็นรหัสที่ดีที่สุดเสมอ



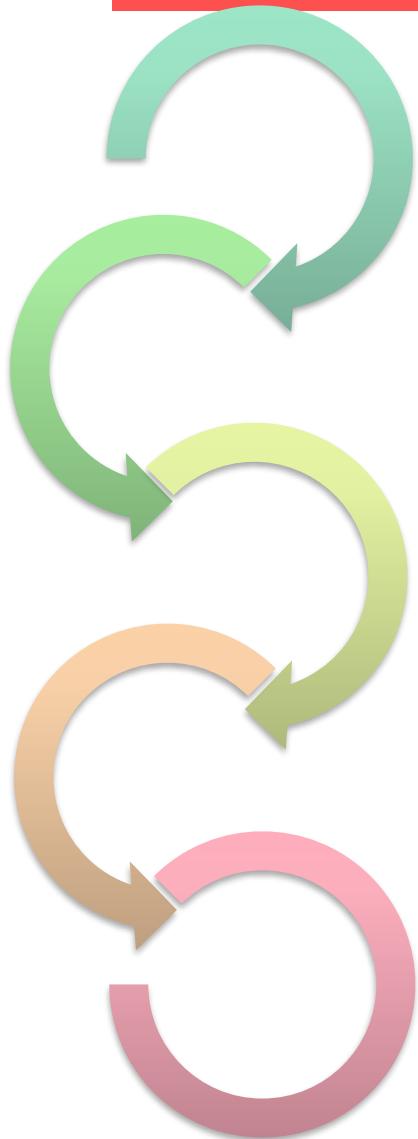
การสร้าง Huffman Coding

Huffman Tree หรือ Binary Tree ของรหัส

สร้างโดยใช้ความถี่ในการเกิด
ของแต่ละตัวอักษรในเอกสาร

ตัวอักษรที่มีความถี่มาก
จะใช้จำนวน Bit น้อยที่สุด

การสร้าง Huffman Coding



หาความถี่ของตัวอักษร (Character) แต่ละตัว
ที่ปรากฏในเอกสารแล้วเก็บไว้ในคิว (Queue)

นำสองตัวที่มีค่าน้อยที่สุดออกจากคิว
ตัวที่มีค่าน้อยสุดอยู่กึ่งซ้าย ส่วนอีกตัวอยู่กึ่งขวา

นำผลรวมของทั้งสองตัวมาเป็น Root และไปเก็บในคิว

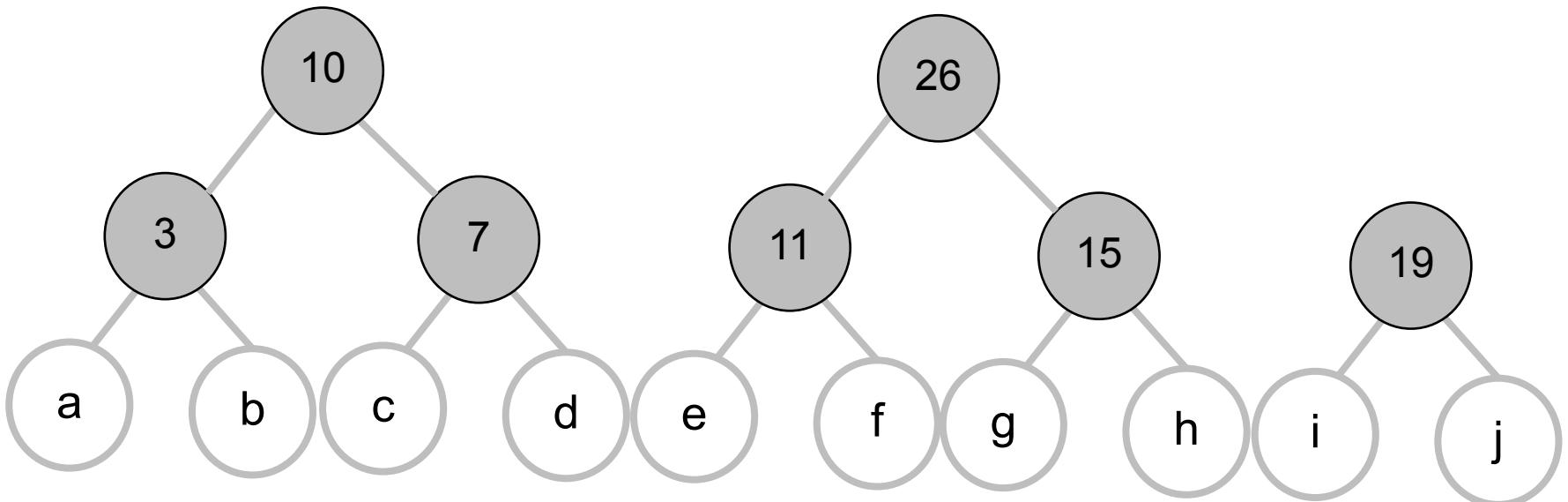
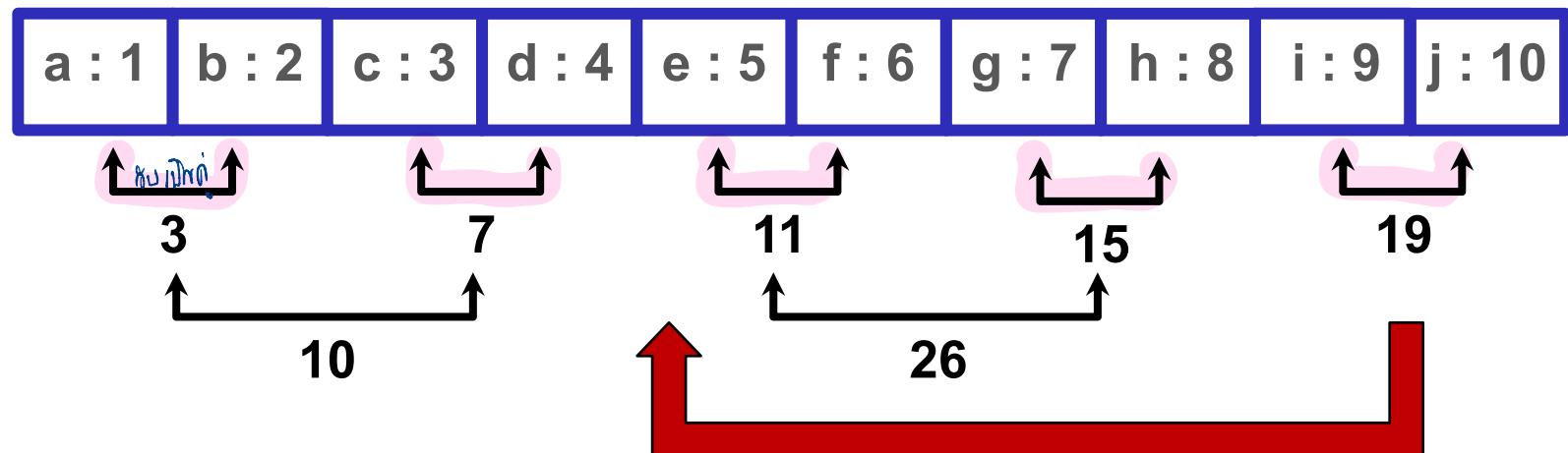
ทำซ้ำข้อ 2 และ 3 จนทั้งหมดรวมเป็น Tree เดียวกัน

Assign ค่าให้กับแต่ละ Character

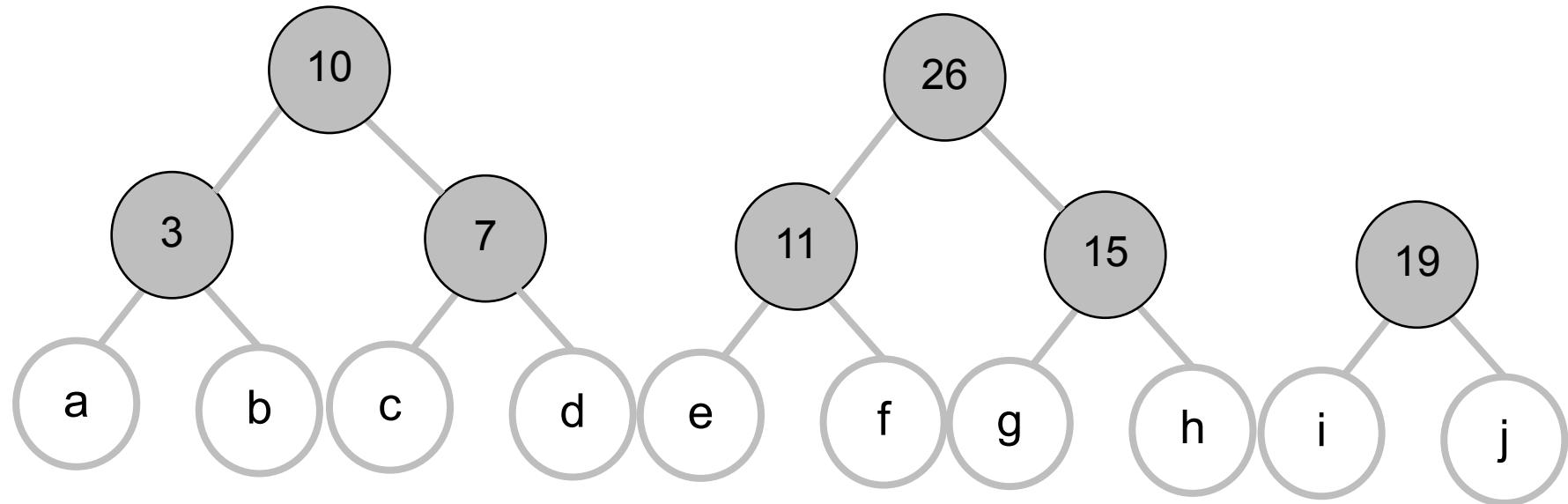
Huffman Coding Example

Case 1

Da return

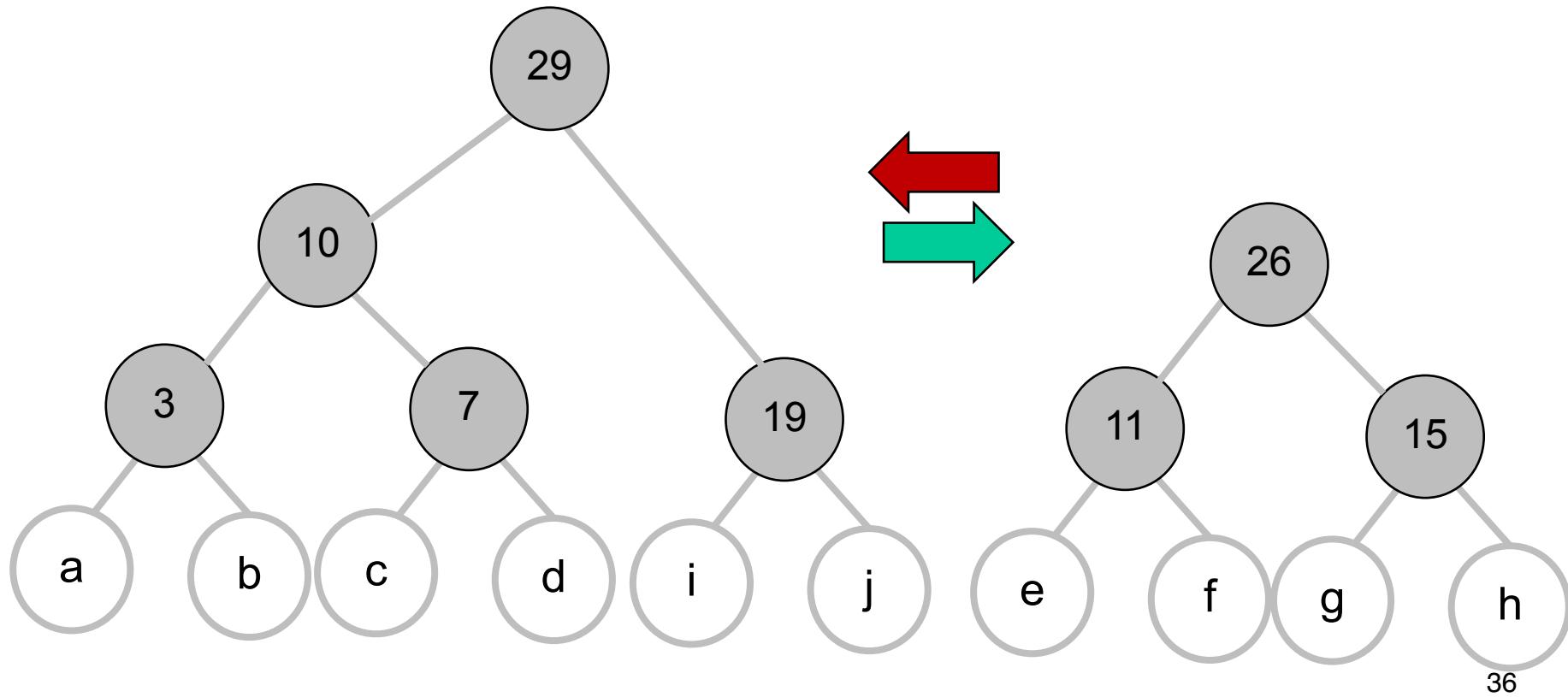


Huffman Coding Example

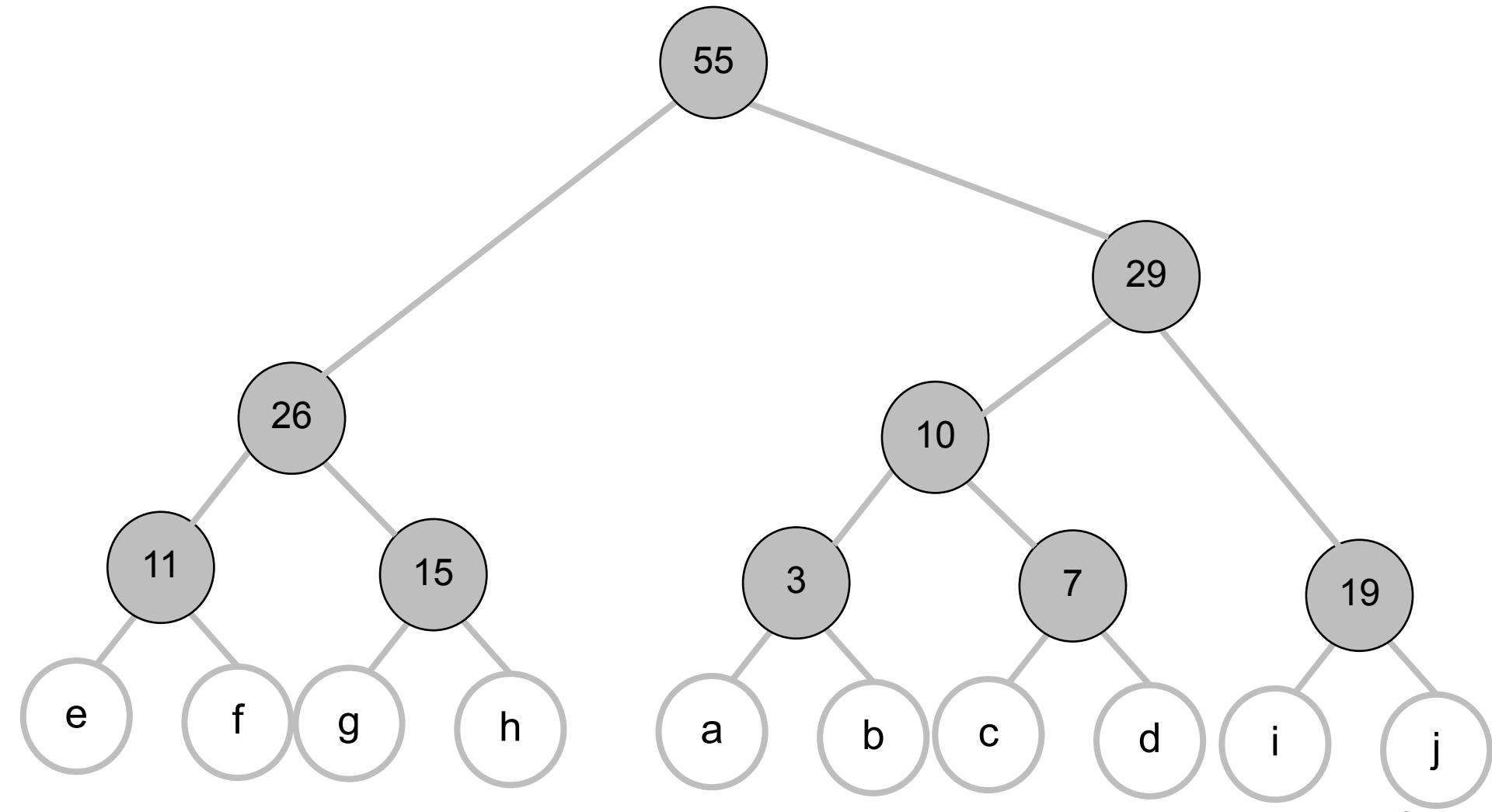


Huffman Coding Example

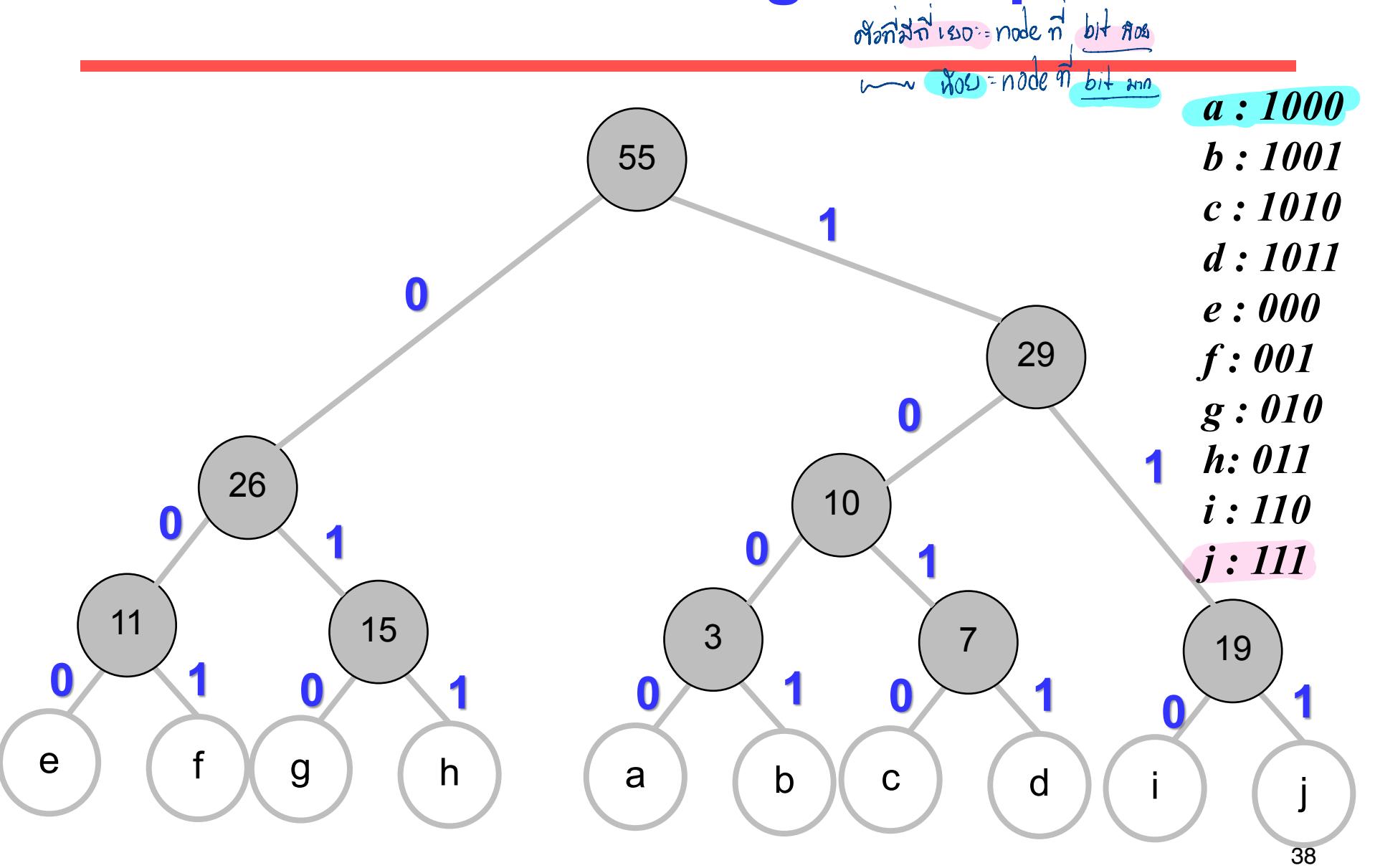
ຂົ້ນ ພວກເຮົາ ລັບໂທ



Huffman Coding Example



Huffman Coding Example

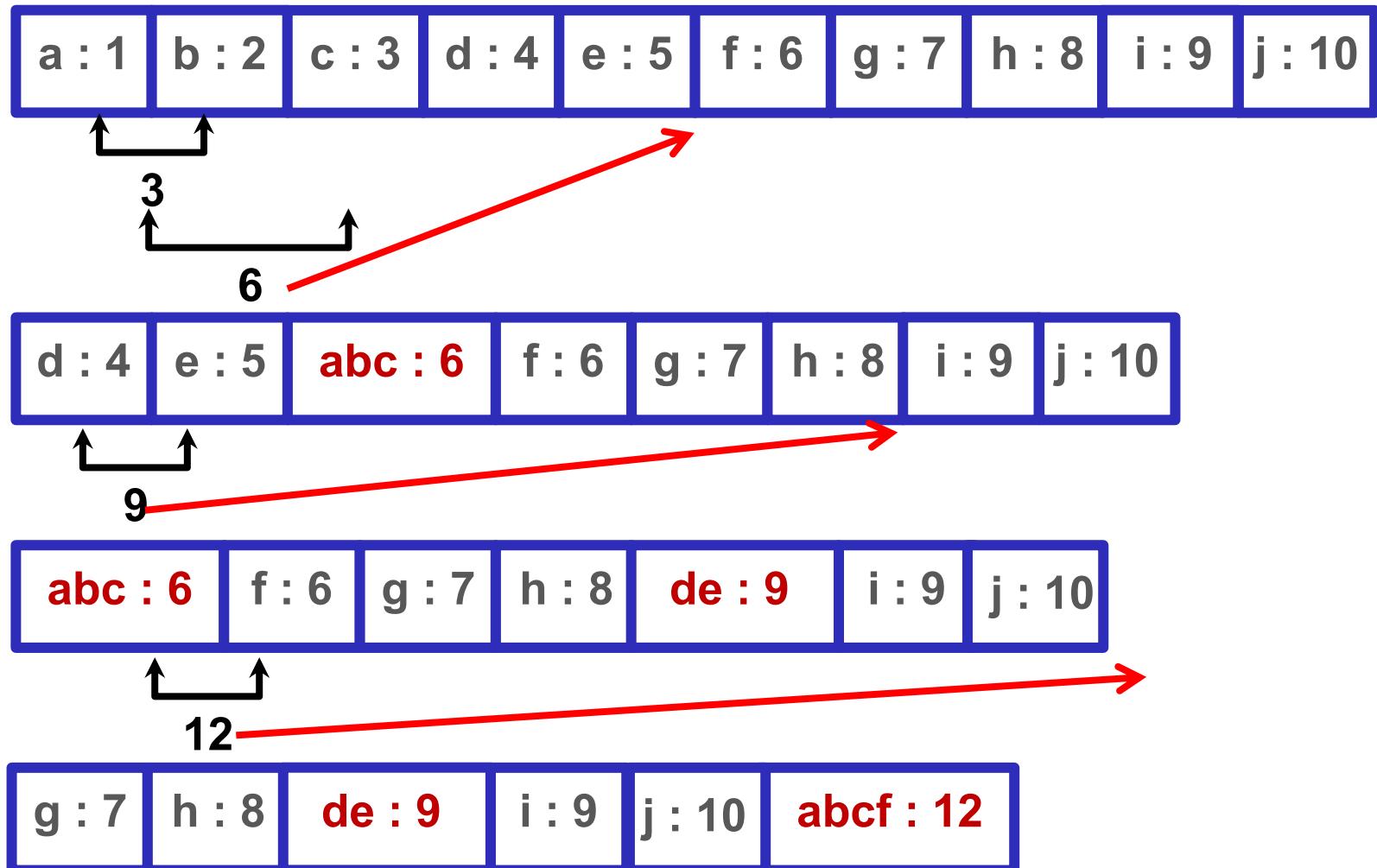


Huffman Coding Example

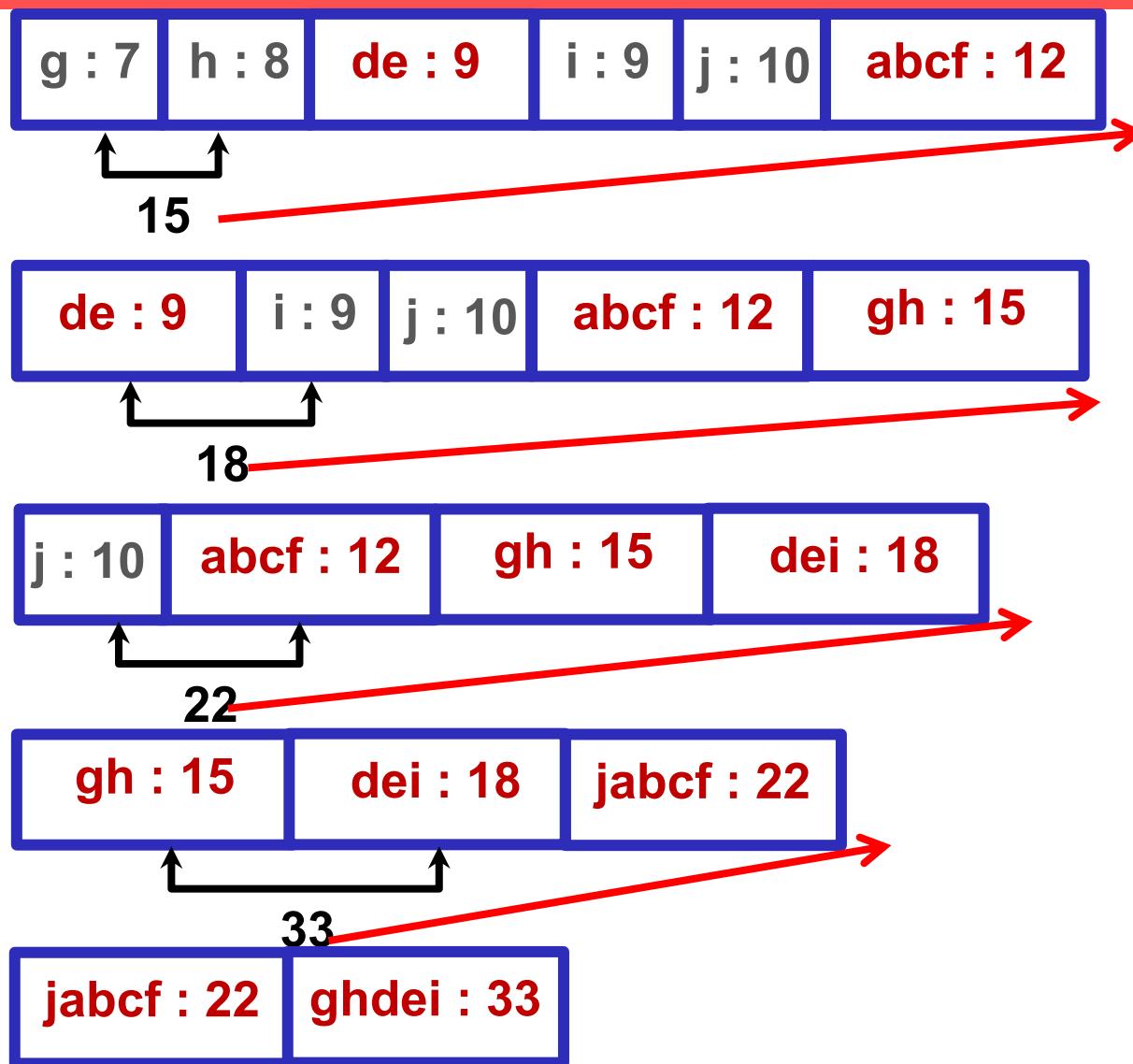
	សម្រាក់ជូន Doubling point	Normal	បង្ហាញអត្ថបទ
a : 1000	4x1	4x1	
b : 1001	4x2	4x2	
c : 1010	4x3	4x3	
d : 1011	4x4	4x4	
e : 000	3x5	4x5	
f : 001	3x6	4x6	
g : 010	3x7	4x7	
h : 011	3x8	4x8	
i : 110	3x9	4x9	
j : 111	3x10	4x10	
$\frac{175}{55} = 3.182$	175 ដែលកើតឡើងពេលបង្កើត	220 ដែលកើតឡើងពេលបង្កើត	
		$\text{Save} = \frac{(220 - 175) * 100}{220} = 20.45\%$	

Huffman Coding Example

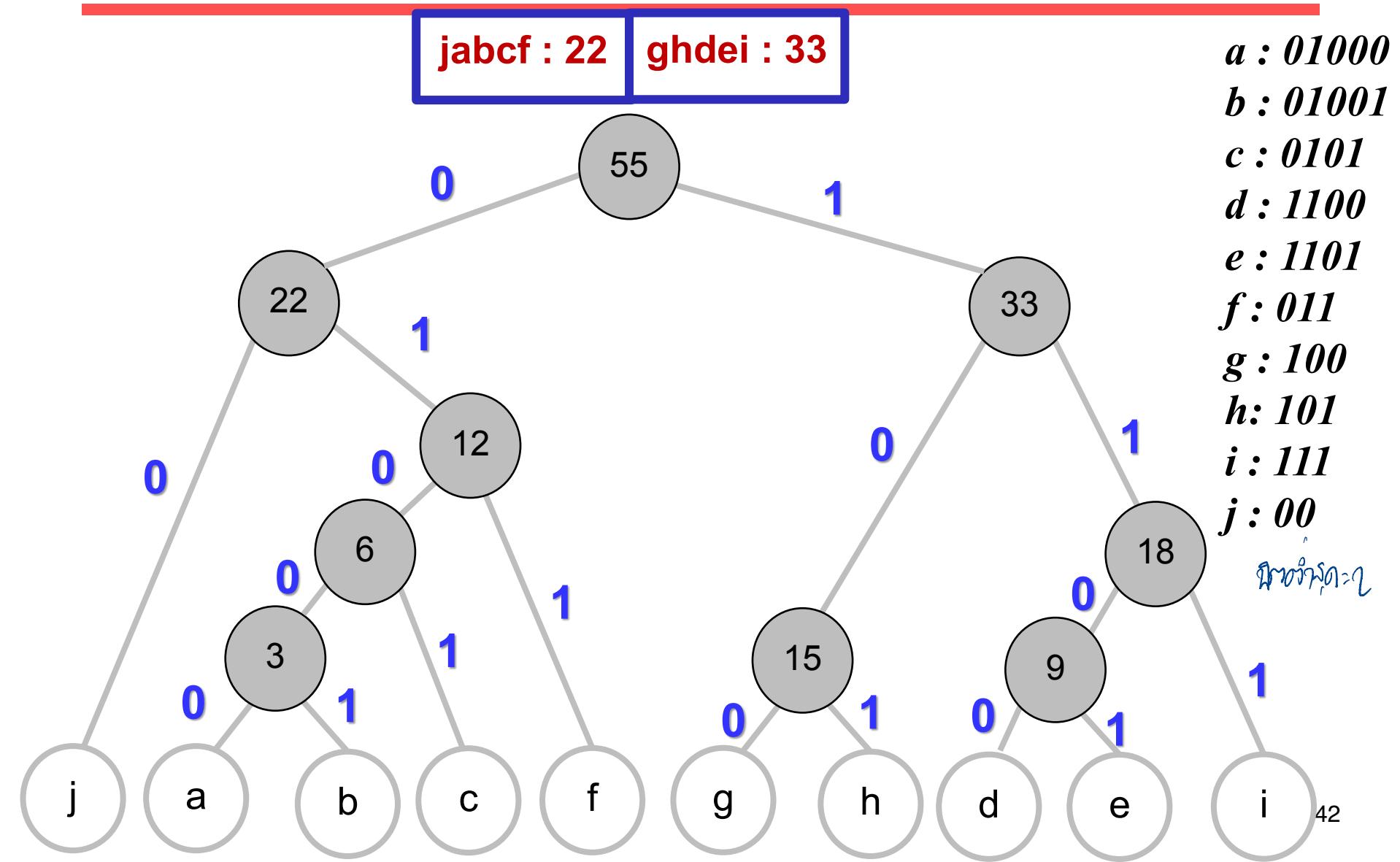
Case 2



Huffman Coding Example



Huffman Coding Example



Huffman Coding Example

ຈົດຕະລິ

ກຳນົມ ດົງກໍາໄງ

		Normal
a : 01000	5x1	4x1
b : 01001	5x2	4x2
c : 0101	4x3	4x3
d : 1100	4x4	4x4
e : 1101	4x5	4x5
f : 011	3x6	4x6
g : 100	3x7	4x7
h : 101	3x8	4x8
i : 111	3x9	4x9
j : 00	2x10	4x10
$\frac{173}{55} = 3.145$	173	220

$$\text{Save} = \frac{(220 - 173) * 100}{220} = 21.36\%$$

Huffman Coding Example

หาความถี่ของตัวอักษรแต่ละตัว

CHARACTER	ความถี่ที่ปรากฏ
a	5000
d	10000
c	20000
b	31000
e	34000

จากนั้นนำมาใส่ในคิว Queue

a : 5000

d : 10000

c : 20000

b : 31000

e : 34000

Huffman Coding Example

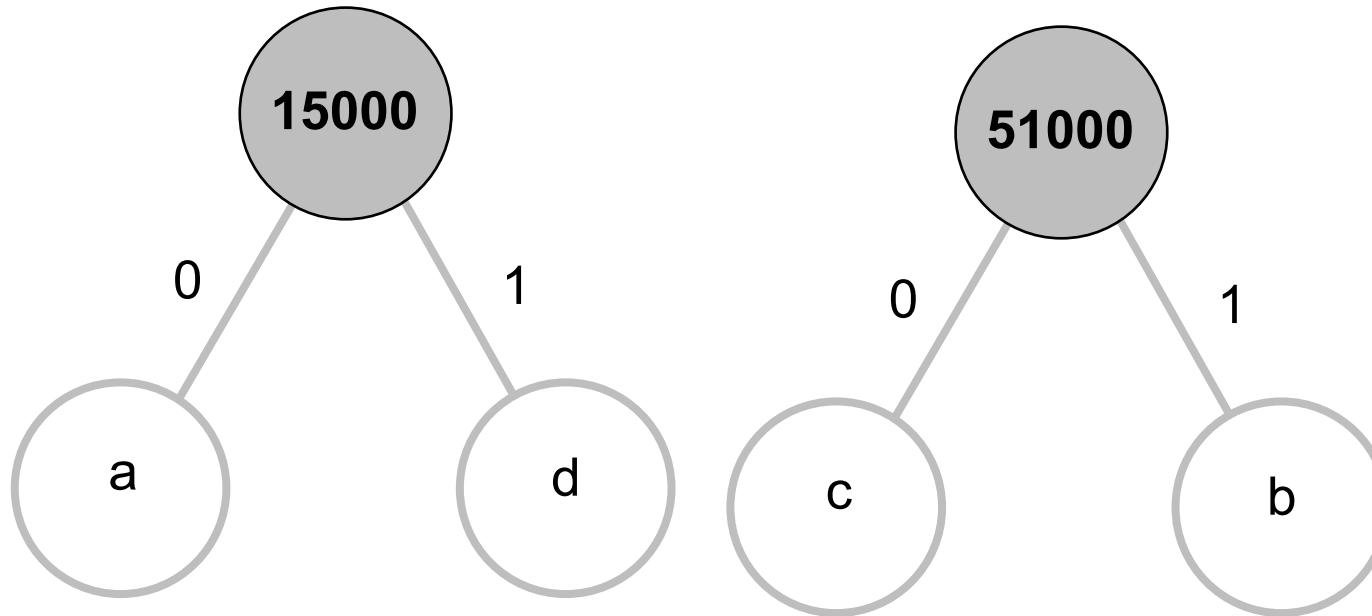
a : 5000

d : 10000

c : 20000

b : 31000

e : 34000

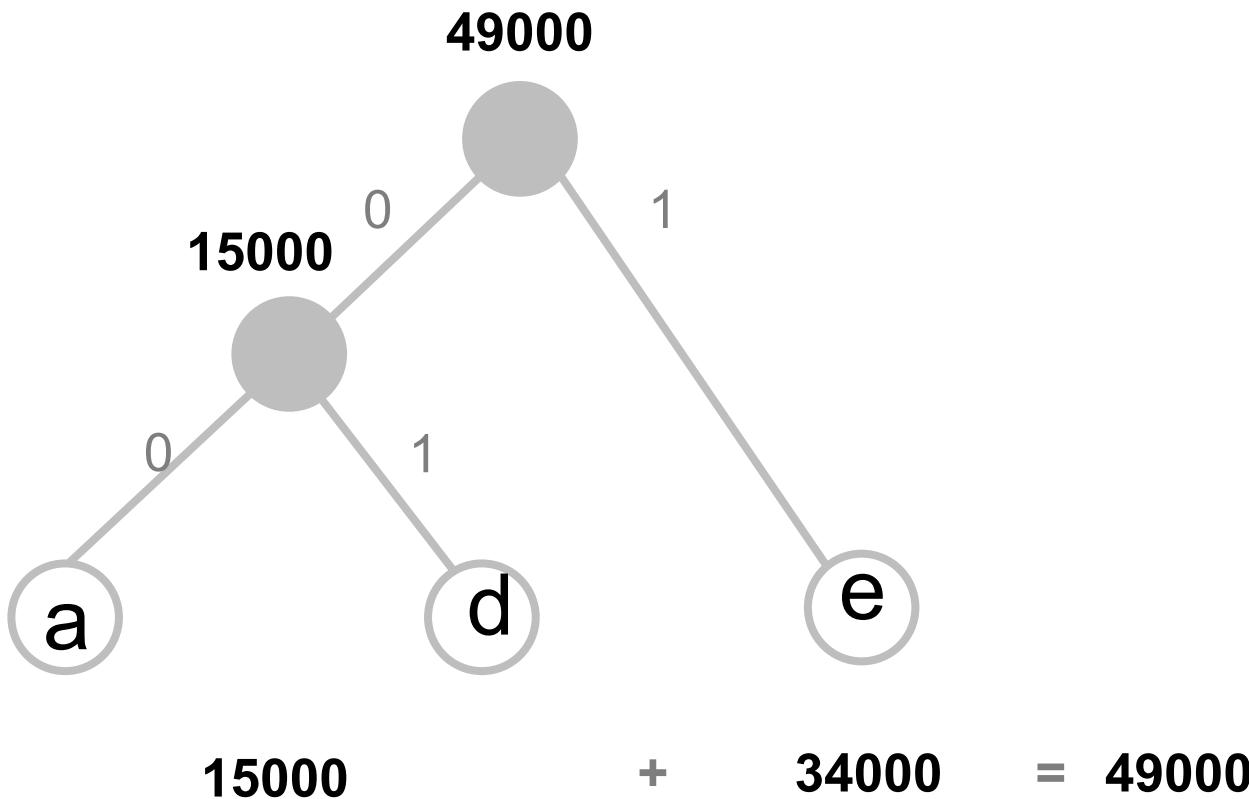


Huffman Coding Example

ad : 15000

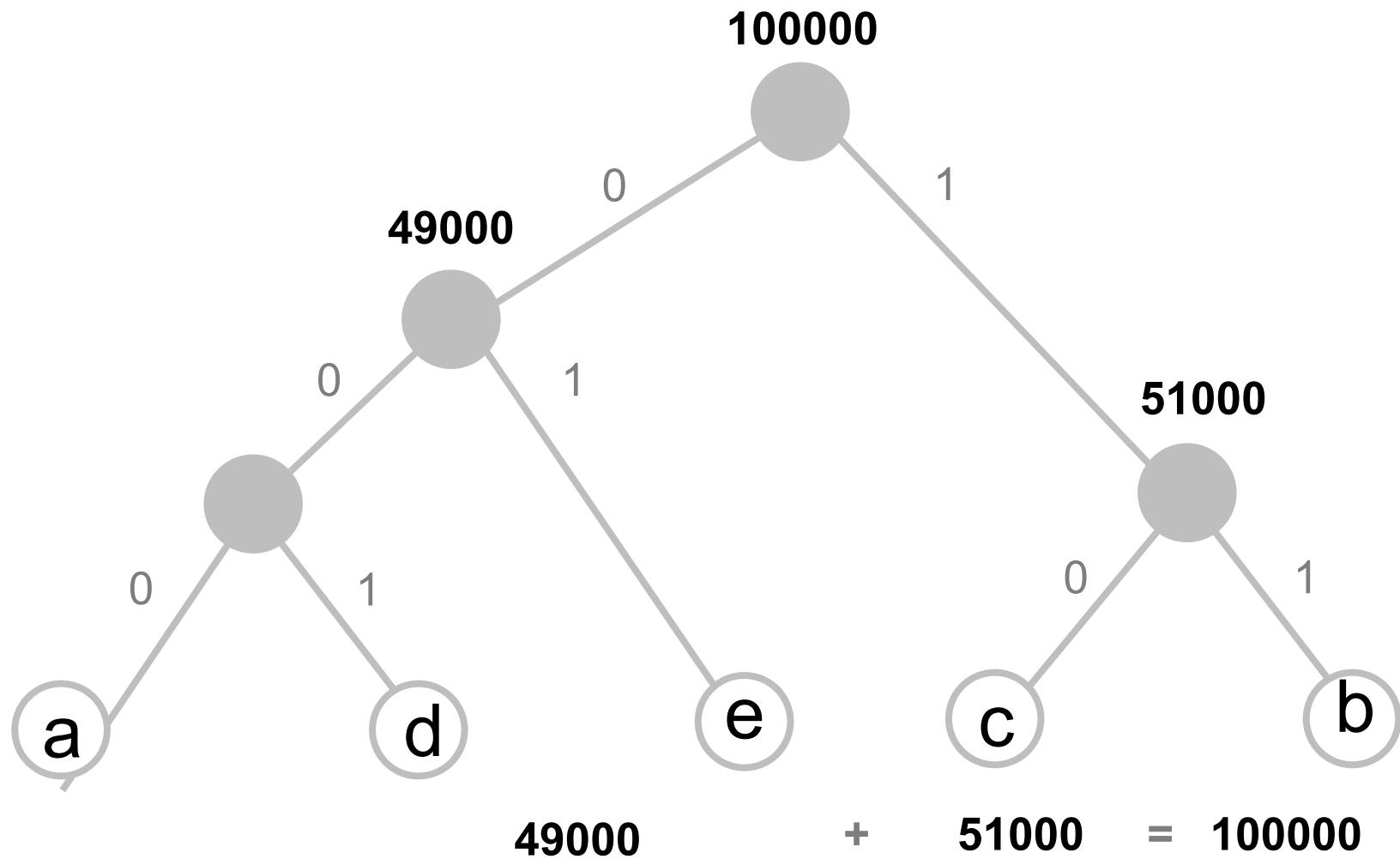
e : 34000

cb : 51000



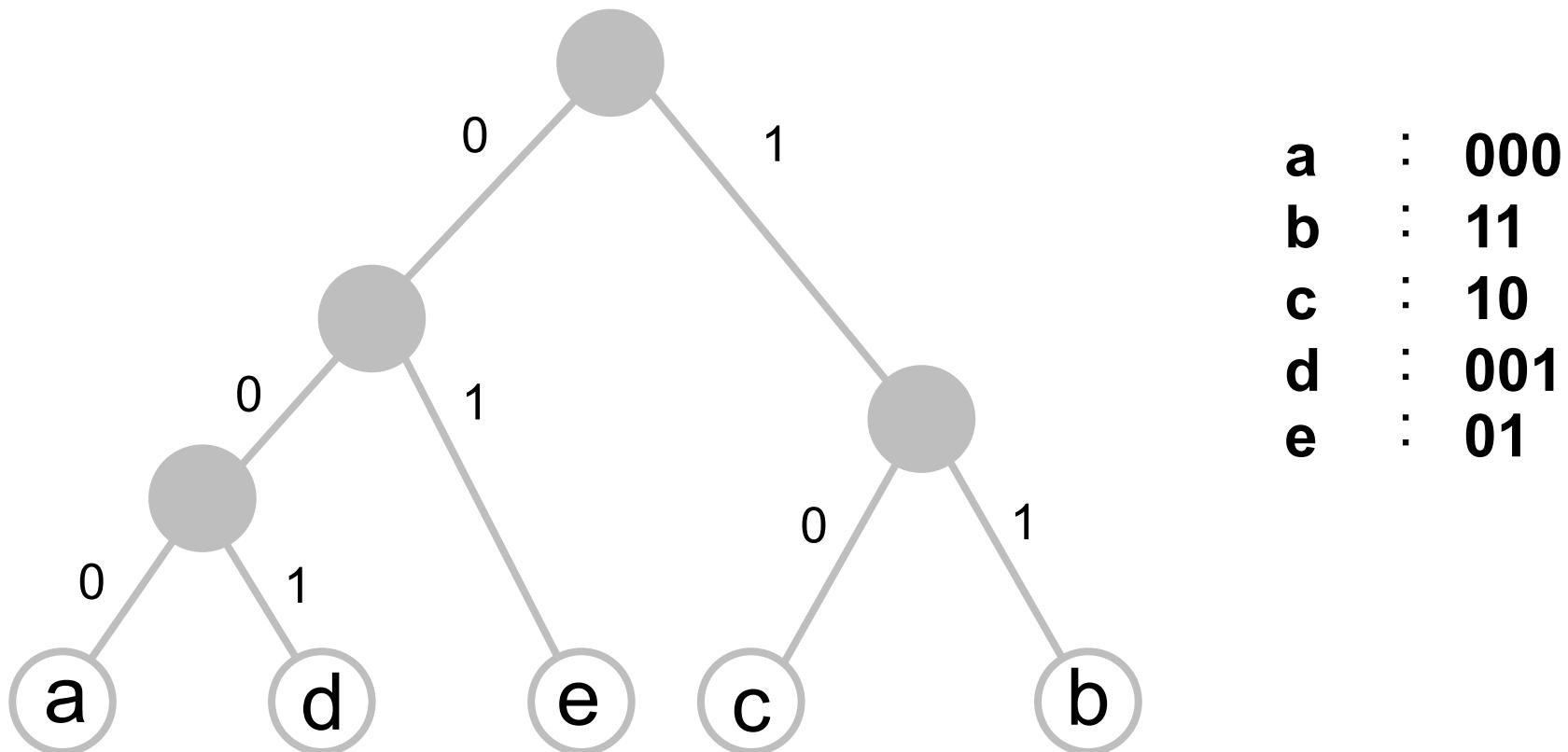
Huffman Coding Example

adecb : 100000



Huffman Coding Example

กำหนดค่าให้กับแต่ละตัวอักษร ได้เป็น Code ของแต่ละตัว

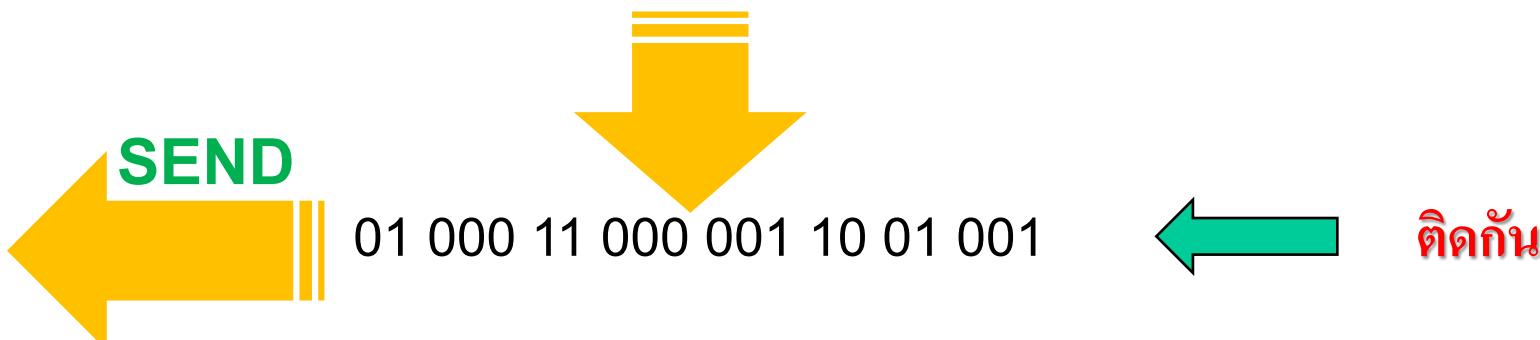


Encoder

ENCODER

a	:	000
b	:	11
c	:	10
d	:	001
e	:	01

e a b a d c e d



Huffman Code

Decoder

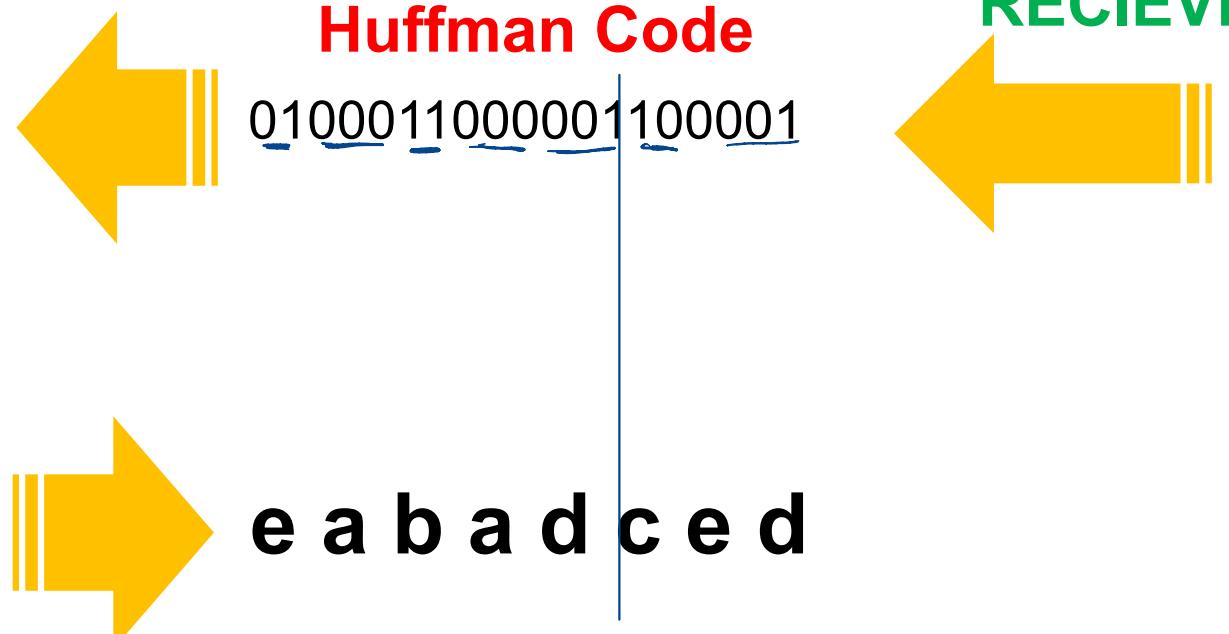
DECODER

000	:	a
11	:	b
10	:	c
001	:	d
01	:	e

Huffman Code

010001100001100001

RECEIVE



การเข้ารหัสโดยไม่ใช้ Huffman

จากตัวอย่างข้างต้นมีตัวอักษรที่ปรากฏทั้งหมด 5 ตัว
ซึ่งถ้าหากทำการเข้ารหัสโดยไม่ได้ใช้ Huffman Code
จะต้องแทนค่าของแต่ละตัวอักษรเป็น 3 บิต ดังนี้

a :	000
b :	001
c :	010
d :	011
e :	100

e a b a d c e d



100 000 001 000 011 010 100 011

การเข้ารหัสโดยไม่ใช้ Huffman

Character	Frequency	Probability	Binary Code	Code Length
a	5000	0.05	000	3
b	31000	0.31	001	3
c	20000	0.20	010	3
d	10000	0.10	011	3
e	34000	0.34	100	3

จำนวนบิตที่ใช้ในการแทนตัวอักษร

$$\frac{1}{f(T)} \times \sum_{i=0}^{i=n} d(i) * f(i)$$

$$= \frac{(5000 \times 3) + (31000 \times 3) + (20000 \times 3) + (10000 \times 3) + (34000 \times 3)}{100000} = 3$$

$$\text{หรือ } =(0.05)*3 + (0.31)*3 + (0.20)*3 + (0.10)*3 + (0.34)*3 = 3 \text{ Bit}$$

การเข้ารหัสโดยใช้ Huffman

a	:	000
b	:	11
c	:	10
d	:	001
e	:	01

e a b a d c e d



01000110000011001001

การเข้ารหัสโดยใช้ Huffman

Character	Frequency	Probability	Binary Code	Code Length
a	5000	0.05	000	3
b	31000	0.31	11	2
c	20000	0.20	10	2
d	10000	0.10	001	3
e	34000	0.34	01	2

จำนวนบิตที่ใช้ในการแทนตัวอักษร

$$\frac{1}{f(T)} \times \sum_{i=0}^{i=n} d(i) * f(i)$$

$$= \frac{(5000 \times 3) + (31000 \times 2) + (20000 \times 2) + (10000 \times 3) + (34000 \times 2)}{100000} = 2.15$$

↑ ต่อไปนี้เป็นสูตรคำนวณ

หรือ $= (0.05)*3 + (0.31)*2 + (0.2)*2 + (0.1)*3 + (0.34)*2 = 2.15 \text{ Bit}$

สรุป

e a b a d c e d

$\sum \text{Probability} * \text{CodeLength}$

การเข้ารหัสโดยไม่ใช้ Huffman

3 Bit

การเข้ารหัสโดยใช้ Huffman

2.15 Bit

Huffman Coding Example

		Normal
a : 000	3x5000	3x5000
b : 11	2x31000	3x31000
c : 10	2x20000	3x20000
d : 001	3x10000	3x10000
e : 01	2x34000	3x34000
	215000	300000

$$\text{Save} = \frac{(300000 - 215000) * 100}{300000} = 28.33\%$$

$$\text{Save} = \frac{(3 - 2.15) * 100}{3} = 28.33\%$$

Text Compression

ព្រមទាំង

- Dictionary-based
 - Ziv-Lempel

Ziv Lempel Welch

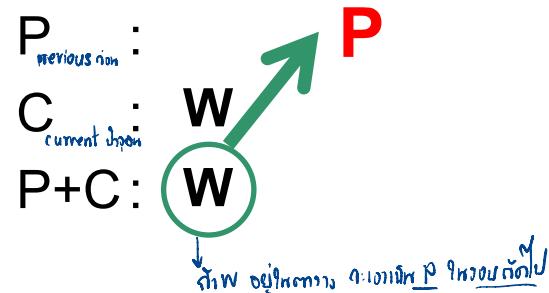
- Ziv-Lempel Coding หรือ Lempel-Ziv-Welch หรือ LZW
- คิดค้นโดย Abraham Lempel และ Jacob Ziv ในปี 1978
- ได้รับการพัฒนาโดย Terry Welch ในปี 1984
- เป็นอัลกอริธึมสำหรับการบีบอัดข้อมูล
- เป็นการบีบอัดที่ให้ผล throughput ที่ดี
- นิยมใช้กันอย่างแพร่หลายใน Unix และ ถูกใช้ในการบีบอัด รูปภาพ (GIF)

Ziv Lempel Welch (example)

Dictionary

1	A
2	B
3	W

Encode Text : WABBAWABBA



Ziv Lempel Welch (example)

Dictionary

1	A
2	B
3	W
4	WA

Encode Text : W A B B A W A B B A

P : W
C : W, A
P+C: W, WA

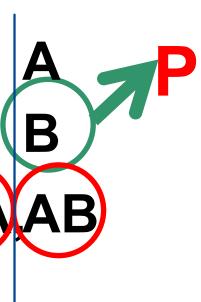
Ziv Lempel Welch (example)

Dictionary

1	A
2	B
3	W
4	WA
5	AB

Encode Text : WABBAWABBA

P : W, A
C : W,A,
P+C: W,WAAB



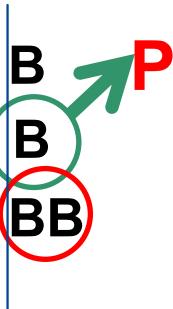
Ziv Lempel Welch (example)

Dictionary

1	A
2	B
3	W
4	WA
5	AB
6	BB

Encode Text : WABBAWABBA

P : W, A, B
C : W, A, B, B
P+C: W, WA, AB, BB



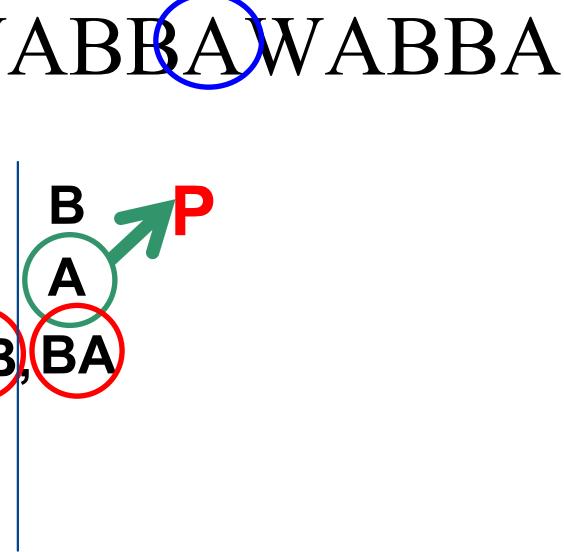
Ziv Lempel Welch (example)

Dictionary

1	A
2	B
3	W
4	WA
5	AB
6	BB
7	BA

Encode Text : WABBAWABBA

P : W, A, B,
C : W, A, B, B,
P+C: W, WA, AB, BB, BA



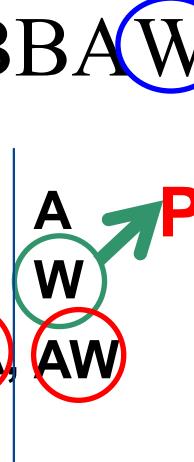
Ziv Lempel Welch (example)

Dictionary

1	A
2	B
3	W
4	WA
5	AB
6	BB
7	BA
8	AW

Encode Text : WABBAWABBA

P : W, A, B, B,
C : W, A, B, B, A,
P+C: W, WA, AB, BB, BA, AW



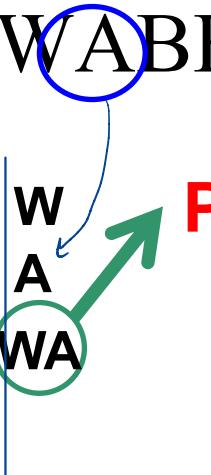
Ziv Lempel Welch (example)

Dictionary

1	A
2	B
3	W
4	WA
5	AB
6	BB
7	BA
8	AW

Encode Text : WABBAWABBA

P : W, A, B, B, A, W, A
C : W,A, B, B, A, W,
P+C: W, WA, AB, BB, BA, AW, WA



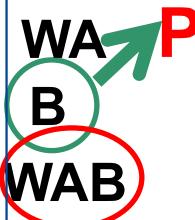
Ziv Lempel Welch (example)

Dictionary

1	A
2	B
3	W
4	WA
5	AB
6	BB
7	BA
8	AW
9	WAB

Encode Text : WABB**A**W**AB**BA

P : W, A, B, B, A, W,
C : W, A, B, B, A, W, A,
P+C: W, WA, AB, BB, BA, AW, WA, WAB



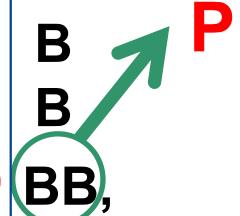
Ziv Lempel Welch (example)

Dictionary

1	A
2	B
3	W
4	WA
5	AB
6	BB
7	BA
8	AW
9	WAB

Encode Text : WABB**A**WAB**B**A

P : W, A, B, B, A, W, WA,
C : W, A, B, B, A, W, A, B,
P+C: W, **WA**, **AB**, **BB**, **BA**, **AW**, WA, **WAB**, BB,



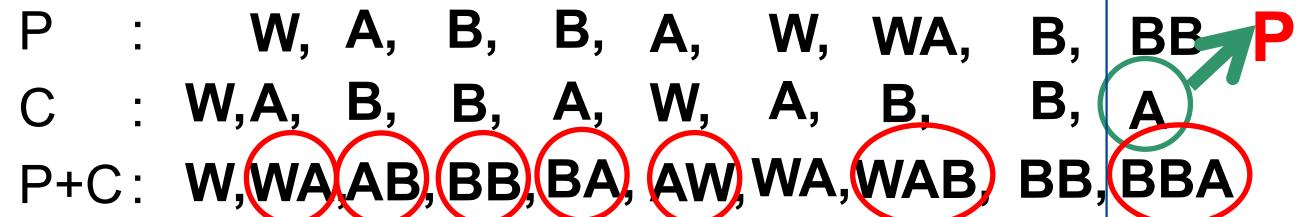
Ziv Lempel Welch (example)

Dictionary

1	A
2	B
3	W
4	WA
5	AB
6	BB
7	BA
8	AW
9	WAB
10	BBA

Encode Text : WABBAWABBA

P : W, A, B, B, A, W, WA, B, BB, P
C : W, A, B, B, A, W, A, B, B, A
P+C: W, WA, AB, BB, BA, AW, WA, WAB, BB, BBA



Ziv Lempel Welch (example)

Dictionary

1	A
2	B
3	W
4	WA
5	AB
6	BB
7	BA
8	AW
9	WAB
10	BBA

Encode Text : WABBAWABBA

↓
inclusion

P : W, A, B, B, A, W, WA, B, BB, A
C : W, A, B, B, A, W, A, B, B, A
P+C: W, **WA**, **AB**, **BB**, **BA**, **AW**, **WA**, **WAB**, **BB**, **BBA**

Ziv Lempel Welch (example)

Dictionary

1	A
2	B
3	W
4	WA
5	AB
6	BB
7	BA
8	AW
9	WAB
10	BBA

Encode Text : WABBAWABBA



Encode(output) : 3 1 2 2 1 4 6 1

start + dictionary (non-injective)

Ziv Lempel Welch (example)

Dictionary

1	A
2	B
3	W

Decode Text : 3 1 2 2 1 4 6 1

Pw : 3
Cw : 3

StrP :
StrC : W 2

P :
C :
P+C:

Output : W

Ziv Lempel Welch (example)

Dictionary

1	A
2	B
3	W
4	WA

Decode Text : 3 1 2 2 1 4 6 1

Pw : 3
Cw : 1

StrP : W
StrC : A

P : W
C : A
P+C: WA

Output : WA

Ziv Lempel Welch (example)

Dictionary

1	A
2	B
3	W
4	WA
5	AB

Decode Text : 3 1 2 2 1 4 6 1

Pw : 1
Cw : 2

StrP : A
StrC : B

P : A
C : B
P+C : AB

Output : WAB

Ziv Lempel Welch (example)

Dictionary

1	A
2	B
3	W
4	WA
5	AB
6	BB

Decode Text : 3 1 2 2 1 4 6 1

Pw : 2

Cw : 2

StrP : B

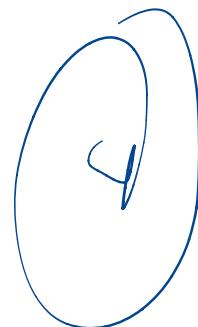
StrC : B

P : B

C : B

P+C: BB

Output : WABB



Ziv Lempel Welch (example)

Dictionary

1	A
2	B
3	W
4	WA
5	AB
6	BB
7	BA

Decode Text : 3 1 2 2 1 4 6 1

Pw : 2

Cw : 1

StrP :

B

StrC :

A

P : B

C : A

P+C : BA

Output : WABBA



Ziv Lempel Welch (example)

Dictionary

1	A
2	B
3	W
4	WA
5	AB
6	BB
7	BA
8	AW

Decode Text : 3 1 2 2 1 4 6 1

Pw : 1

Cw : 4

StrP :

A

StrC :

WA

P : A

C : W

P+C: AW

Output : **WABBAWA**



Ziv Lempel Welch (example)

Dictionary

1	A
2	B
3	W
4	WA
5	AB
6	BB
7	BA
8	AW
9	WAB

Decode Text : 3 1 2 2 1 4 6 1

Pw : 4

Cw : 6

StrP :

WA

StrC :

BB

P : WA

C : B

P+C: WAB

Output : WABB AWABB

Ziv Lempel Welch (example)

Dictionary

1	A
2	B
3	W
4	WA
5	AB
6	BB
7	BA
8	AW
9	WAB
10	BBA

Decode Text : 3 1 2 2 1 4 6 1

Pw : 6

Cw : 1

StrP :

BB

StrC :

A

P : BB

C : A

P+C: BBA

Output : WABBAWABBA