جامعة القاهرة
كلية الحاسبات
و المعلومات

Variable Length Coding

Prof. Khaled Mostafa
khaledms@fci-cu.edu.eg

Variable Length Coding (VLC)

abaacaadaa *(10 Symbols)*

P(a)=7/10, P(b)=1/10, P(c)=1/10, P(d)=1/10

Binary System assigns 2 Bits for Each Symbol

1

Symbol	Count	Code	Code Length	Total Bits
a	7	00	2	14
b	1	01	2	2
c	1	10	2	2
d	1	11	2	2

00010000100000110000

Symbols Can be stored in 20 Bits

We can Use Variable Length Codes for different Symbols

2

Symbol	Count	Code	Code Length	Total Bits
a	7	0	1	7
b	1	10	2	2
c	1	110	3	3
d	1	111	3	3

010001100011100

Symbols Can be stored In 15 Bits

Prof. Khaled Mostafa
khaledms@fci-cu.edu.eg

Variable Length Coding (VLC)

abaacaadaa *(10 Symbols)*

P(a)=7/10, P(b)=1/10, P(c)=1/10, P(d)=1/10

Another Variable Length Codes for different Symbols

3

Symbol	Count	Code	Code Length	Total Bits
a	7	0	1	7
b	1	1	1	1
c	1	01	2	2
d	1	00	2	2

010001000000

Symbols Can be stored in 12 Bits

Another Variable Length Codes for different Symbols

4

Symbol	Count	Code	Code Length	Total Bits
a	7	10	2	14
b	1	0	1	1
c	1	111	3	3
d	1	110	3	3

100101011110101101010

Symbols Can be stored In 21 Bits

Prof. Khaled Mostafa
khaledms@fci-cu.edu.eg

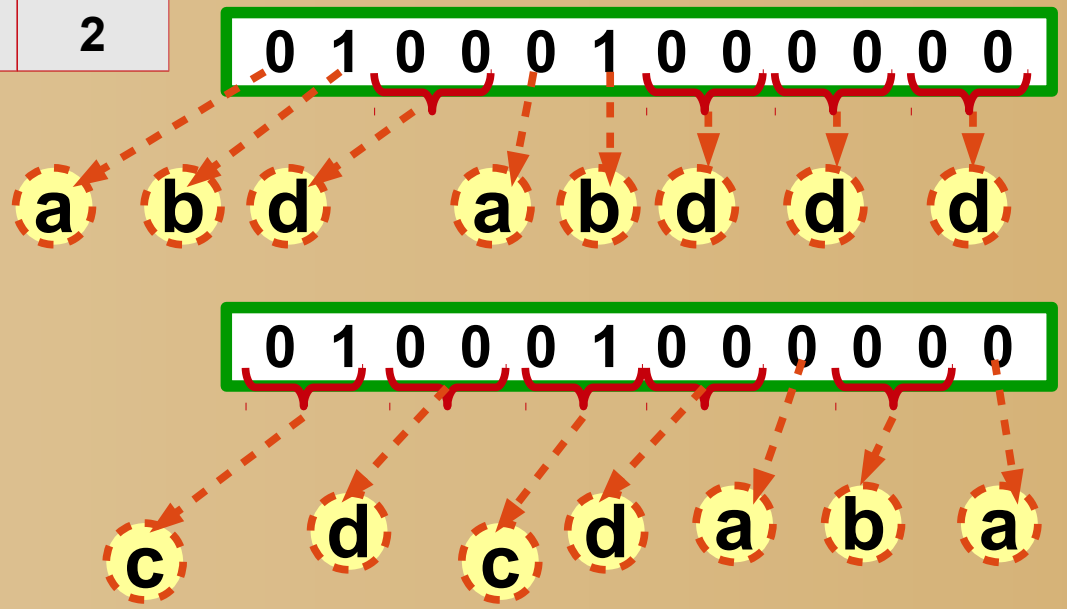
Confusing Codes

abaacaadaa *(10 Symbols)*

P(a)=7/10, P(b)=1/10, P(c)=1/10, P(d)=1/10

3

Symbol	Count	Code	Code Length	Total Bits
a	7	0	1	7
b	1	1	1	1
c	1	01	2	2
d	1	00	2	2



Prof. Khaled Mostafa
khaledms@fci-cu.edu.eg

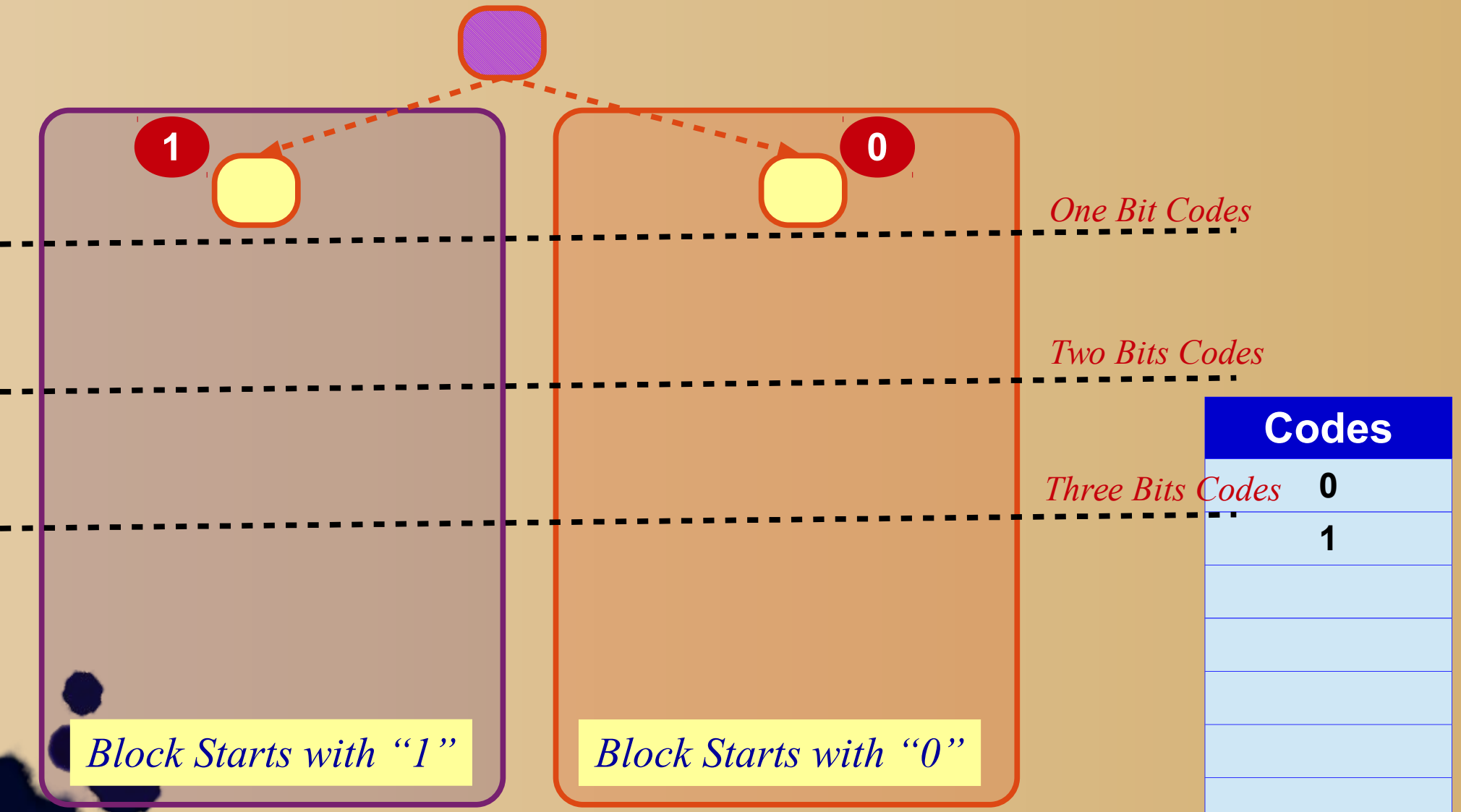
جامعة القاهرة
كلية الحاسبات
و المعلومات

Prefix Conditional Code Generation

No Symbol Code is a prefix of any other Symbol Code

Prof. Khaled Mostafa
khaledms@fci-cu.edu.eg

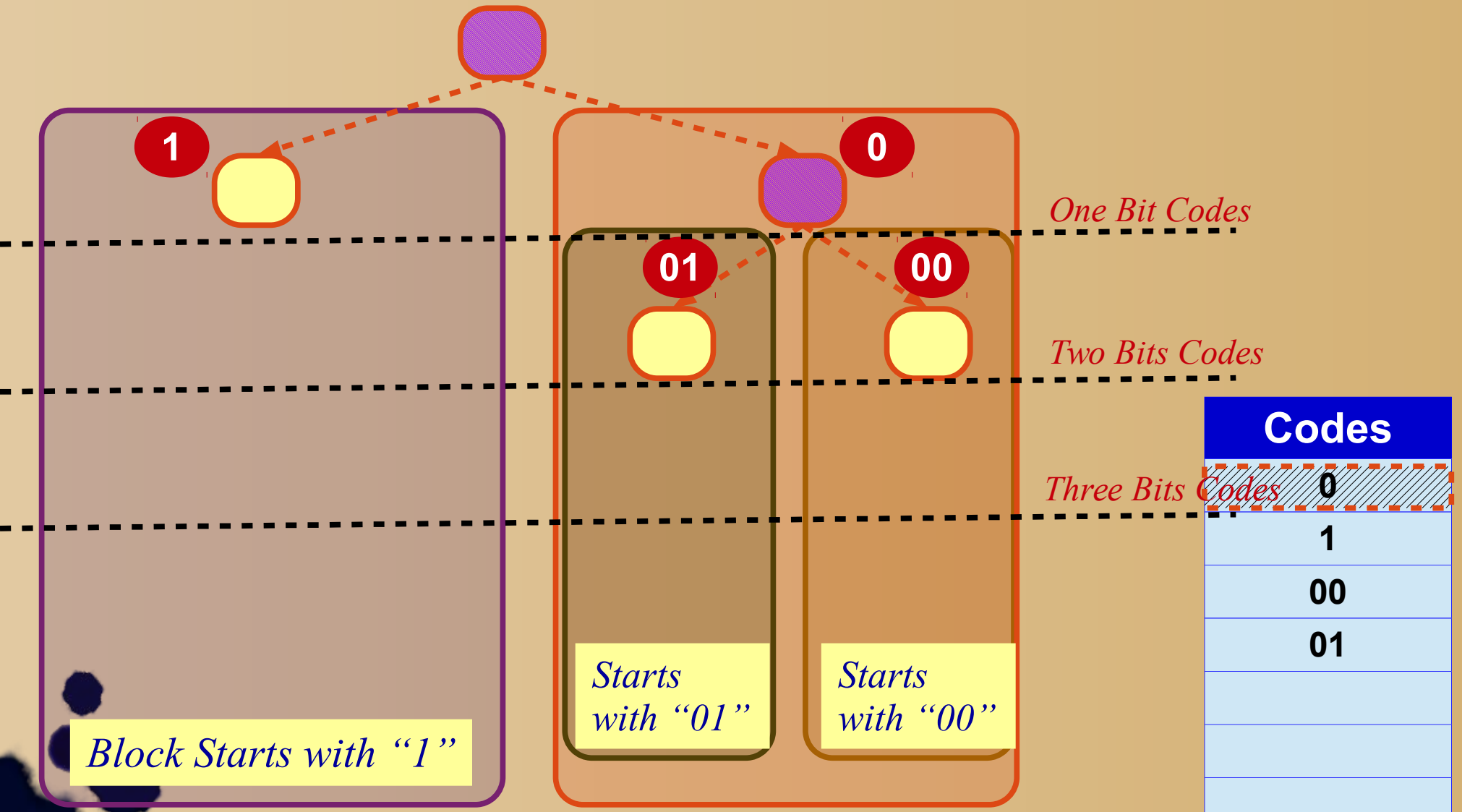
Generation of Prefix Conditional Codes



Codes
0
1

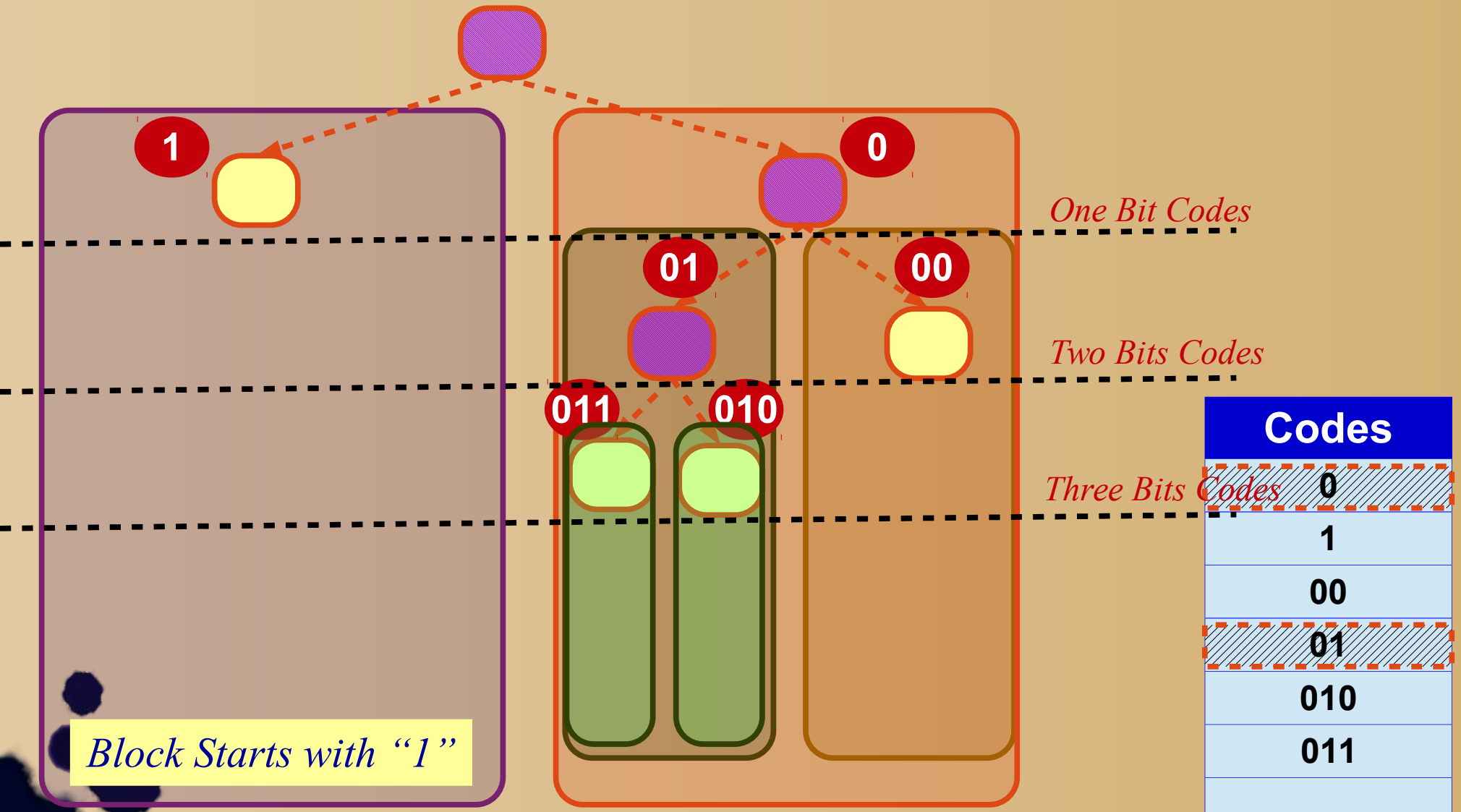
Prof. Khaled Mostafa
khaledms@fci-cu.edu.eg

Generation of Prefix Conditional Codes



Prof. Khaled Mostafa
khaledms@fci-cu.edu.eg

Generation of Prefix Conditional Codes



Codes
0
1
00
01
010
011

Prof. Khaled Mostafa
khaledms@fci-cu.edu.eg

Which of these Codes are Prefix Conditional Codes

Symbol	Code
a	0
b	10
c	110
d	1110
e	11110
f	11111

?

Symbol	Code
a	0
b	11
c	101
d	100
e	111
f	01

?

Symbol	Code
a	001
b	0001
c	10
d	110
e	0000
f	111

?

Prof. Khaled Mostafa
khaledms@fci-cu.edu.eg

جامعة القاهرة
كلية الحاسبات
و المعلومات

Entropy

Prof. Khaled Mostafa
khaledms@fci-cu.edu.eg

Entropy

جامعة القاهرة
كلية الحاسبات
و المعلومات

Entropy is the average amount of information contained in each message received. Here, message stands for an event, Symbol, or character drawn from a distribution or data stream.

Entropy is a measure of *information content:* the number of bits *actually* required to store data.

Entropy is sometimes called a **measure of** *uncertainty* *or* *disorder* *or* *unpredictability* *of information content*

A ***highly predictable*** *sequence contains* ***little actual information***

Prof. Khaled Mostafa
khaledms@fci-cu.edu.eg

Entropy of Source Data

جامعة القاهرة
كلية الحاسبات
و المعلومات

Information content "I"associated with any symbol "S"
is reversely proportional to its probability

$$I(S) = \text{Log}_2 \{1/P(S)\}$$

Information Content "I" associated with ALL Symbols (S0, S1, S2, ..Sn)=

$$I(\text{All Symbols}) = \text{Log}_2 \{1/P(S_1)\} + \text{Log}_2 \{1/P(S_2)\} + \text{Log}_2 \{1/P(S_3)\} + \ldots + \text{Log}_2 \{1/P(S_n)\}$$

$$I(All\ Symbols) = \sum_{i=0}^{i=n} \log_2 \{1/P(S_i)\}$$

Prof. Khaled Mostafa
khaledms@fci-cu.edu.eg

Entropy of Source Data

Average Information Content "H" associated with ALL Symbols (S0, S1, S2, ..Sn)=

$$H(S)=1/M\ [m_1\ Log_2\ \{1/P(S_1)\} + m_2\ Log_2\ \{1/P(S_2)\} + m_3\ Log_2\ \{1/P(S_3)\} +...+ m_n\ Log_2\ \{1/P(S_n)\}]$$

Where S_1 is repeated m_1 times, S_2, is repeated m_2 times, , … and Total Number of Symbols $M= m_1+m_2+..m_n$

$$H(S)=(m_1/M)\ Log_2\ \{1/P(S_1)\} + (m_2/M)\ Log_2\{1/P(S_2)\} + (m_3/M)\ Log_2\{1/P(S_3)\} +...+ (m_n/M)\ Log_2\{1/P(S_n)\}$$

Prof. Khaled Mostafa
khaledms@fci-cu.edu.eg

Entropy of Source Data

$$H(S)=(P(S_1)\ \mathrm{Log}_2\ \{1/P(S_1)\} + (P(S_2)\ \mathrm{Log}_2\{1/P(S_2)\} + (P(S_3)\ \mathrm{Log}_2\{1/P(S_3)\} +\ldots+ (P(S_n)\ \mathrm{Log}_2\{1/P(S_n)\}$$

$$H(S)=\sum_{i=0}^{i=n} P(S_i)\ \log_2\ \{1/P(S_i)\}$$

Prof. Khaled Mostafa
khaledms@fci-cu.edu.eg

Entropy of Source Data

$$H(S)=\sum_{i=0}^{i=n} P(S_i)\ \log_2 \{1/P(S_i)\}$$

Note:

$\log_2 X = \log_{10} x / \log_{10} 2 = \log_{10} x / 0.301$

Calculate Entropy for the following Data **a b a a c a a d a a**

Symbol	P(S)	Log_2 [1/P(S)]	P(S) * Log_2 [1/P(S)]
a	0.7	0.5145	0.36
b	0.1	3.3219	0.33219
c	0.1	3.3219	0.33219
d	0.1	3.3219	0.33219

Entropy H(S) = 1.35687 Bits / symbol

Prof. Khaled Mostafa
khaledms@fci-cu.edu.eg

Shannon Source Coding Theorem

جامعة القاهرة
كلية الحاسبات
و المعلومات

For a Discrete Memoryless System (Where no Prediction is Allowed) The maximum level of compression can be reached is the Entropy H(S) measured in Bits/ Symbol

Prof. Khaled Mostafa
khaledms@fci-cu.edu.eg

Entropy Calculation Example

جامعة القاهرة
كلية الحاسبات
و المعلومات

abaacaadaa *(10 Symbols)*

P(a)=7/10, P(b)=1/10, P(c)=1/10, P(d)=1/10

$$H(S)=\sum_{i=0}^{i=n} P(S_i)\log_2\{1/P(S_i)\}$$

Note:

$\log_2 X = \log_{10} x / \log_{10} 2 = \log_{10} x / 0.301$

$H(S)=0.7*\log_2(1/0.7)+0.1*\log_2(1/0.1)+0.1*\log_2(1/0.1)+0.1*\log_2(1/0.1)=$

$0.7*0.515+0.1*3.322*3 = 1.375$ Bits / Symbol

the Minimum Memory less compression size that can be reached Is **1.375 bits /symbol**

(e.g. 10 symbols can be stored in 10*1.375=13.75 bits ~=14 bits

Prof. Khaled Mostafa
khaledms@fci-cu.edu.eg

Different Codes for Symbols

abaacaadaa *(10 Symbols)*

P(a)=7/10, P(b)=1/10, P(c)=1/10, P(d)=1/10

Coding 1: Binary System

Symbol	Count	Code	Code Length	Total Bits
a	7	00	2	14
b	1	01	2	2
c	1	10	2	2
d	1	11	2	2

Compressed Total = 20 Bits

Coding 2: Huffman

Symbol	Count	Code	Code Length	Total Bits
a	7	0	1	7
b	1	10	2	2
c	1	110	3	3
d	1	111	3	3

Compressed Total = 15 Bits

Compressed (According to Entropy) = 14 Bits

Prof. Khaled Mostafa
khaledms@fci-cu.edu.eg

Entropy Calculation Example

جامعة القاهرة
كلية الحاسبات
و المعلومات

P(a)=0.17, P(b)=0.22, P(c)=0.15, P(d)=0.14, P(e)=0.3, P(f)=0.02

$$H(S)=\sum_{i=0}^{i=n} P(S_i)\ \log_2 \{1/P(S_i)\}$$

Note:
$\log_2 X = \log_{10} X / \log_{10} 2$

$H(S)=0.17*\log_2(1/0.17) + 0.22*\log_2(1/0.22) + 0.15*\log_2(1/0.15) + 0.14*\log_2(1/0.14) + 0.30*\log_2(1/0.30) + 0.02*\log_2(1/0.02) =$
2.3567 Bits / Symbol

the Minimum Memory less compression size that can be reached is **2.3567 bits /symbol**
(e.g. 100 symbols can be stored in 100*2.3567=235.67 bits ~=236 bits

Prof. Khaled Mostafa
khaledms@fci-cu.edu.eg

جامعة القاهرة
كلية الحاسبات
و المعلومات

Huffman Coding Algorithm

Prof. Khaled Mostafa
khaledms@fci-cu.edu.eg

Huffman Coding Algorithm

- Each symbol is a leave node in a tree
- Combining the two symbols or composite symbols with the least probabilities to form a new parent composite symbols, which has the combined probabilities. Assign a bit 0 and 1 to the two links
- Continue this process till all symbols merged into one root node. For each symbol, the sequence of the 0s and 1s from the root node to the symbol is the code word

Prof. Khaled Mostafa
khaledms@fci-cu.edu.eg

Huffman Coding Example

P(a)=0.17, P(b)=0.22, P(c)=0.15, P(d)=0.14, P(e)=0.3, P(f)=0.02

P(e)= 0.3

P(b)= 0.22

P(a)= 0.17

P(c)= 0.15

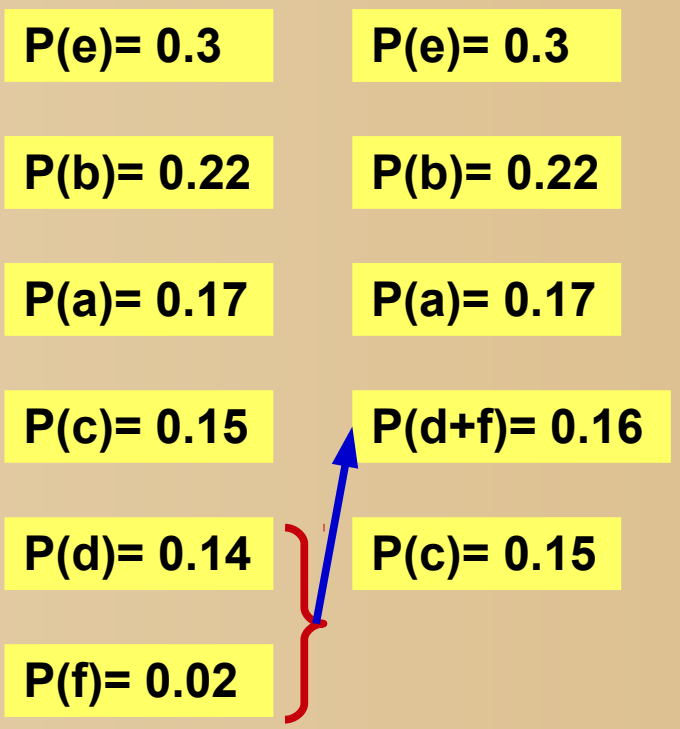
P(d)= 0.14

P(f)= 0.02

Order Symbols According to Their probabilities (Descending)

Prof. Khaled Mostafa
khaledms@fci-cu.edu.eg

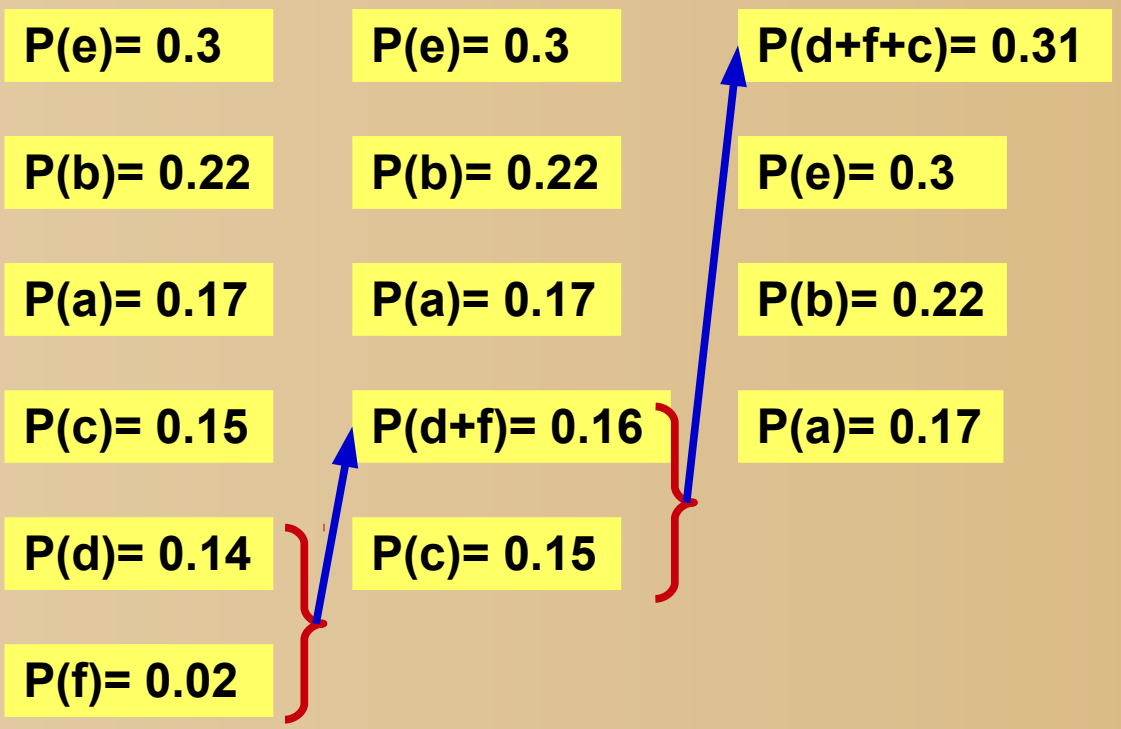
Huffman Coding Example



Combine Last two Symbols (with Lowest Probabilities),
Then reorder the list

Prof. Khaled Mostafa
khaledms@fci-cu.edu.eg

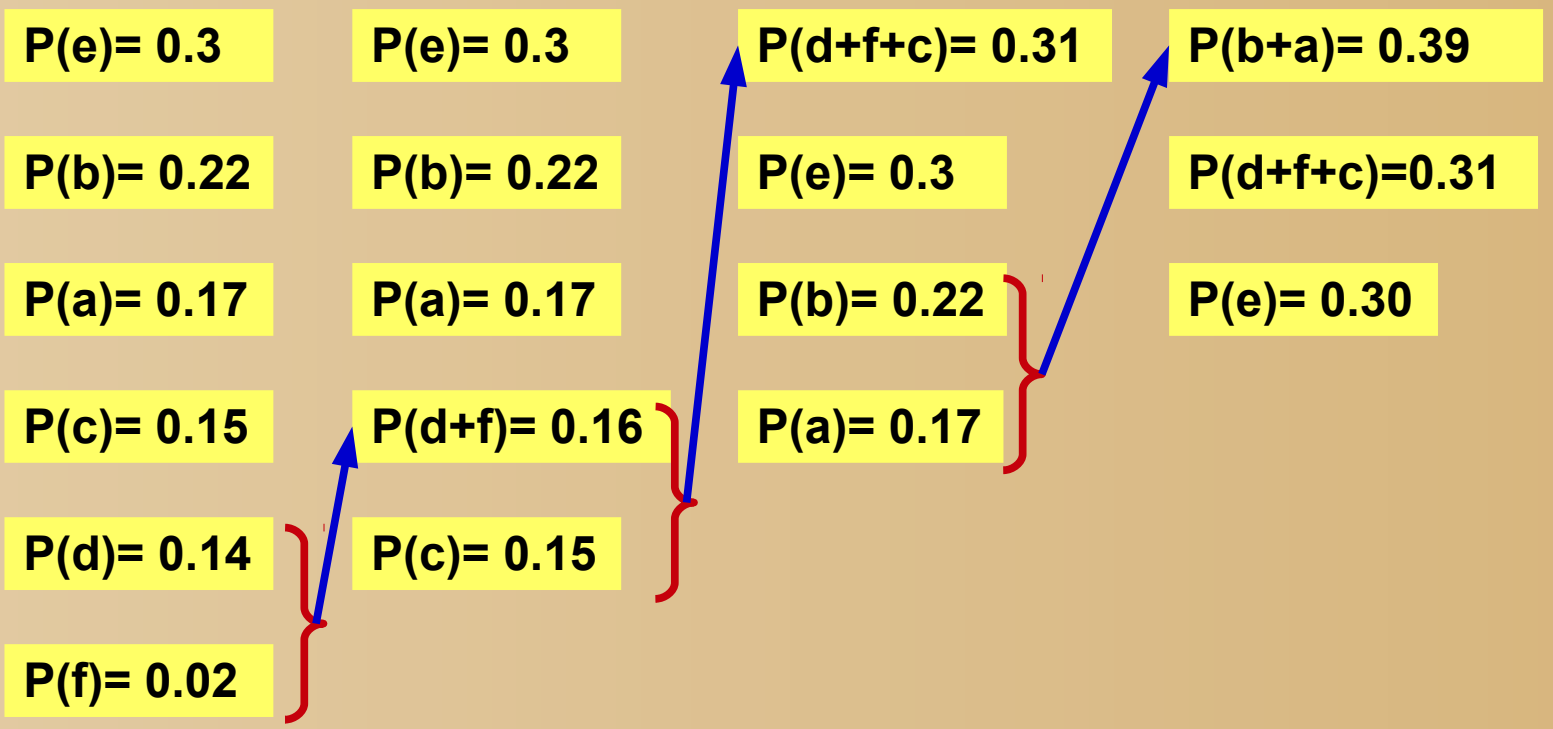
Huffman Coding Example



Combine Last two Symbols (with Lowest Probabilities),
Then reorder the list

Prof. Khaled Mostafa
khaledms@fci-cu.edu.eg

Huffman Coding Example

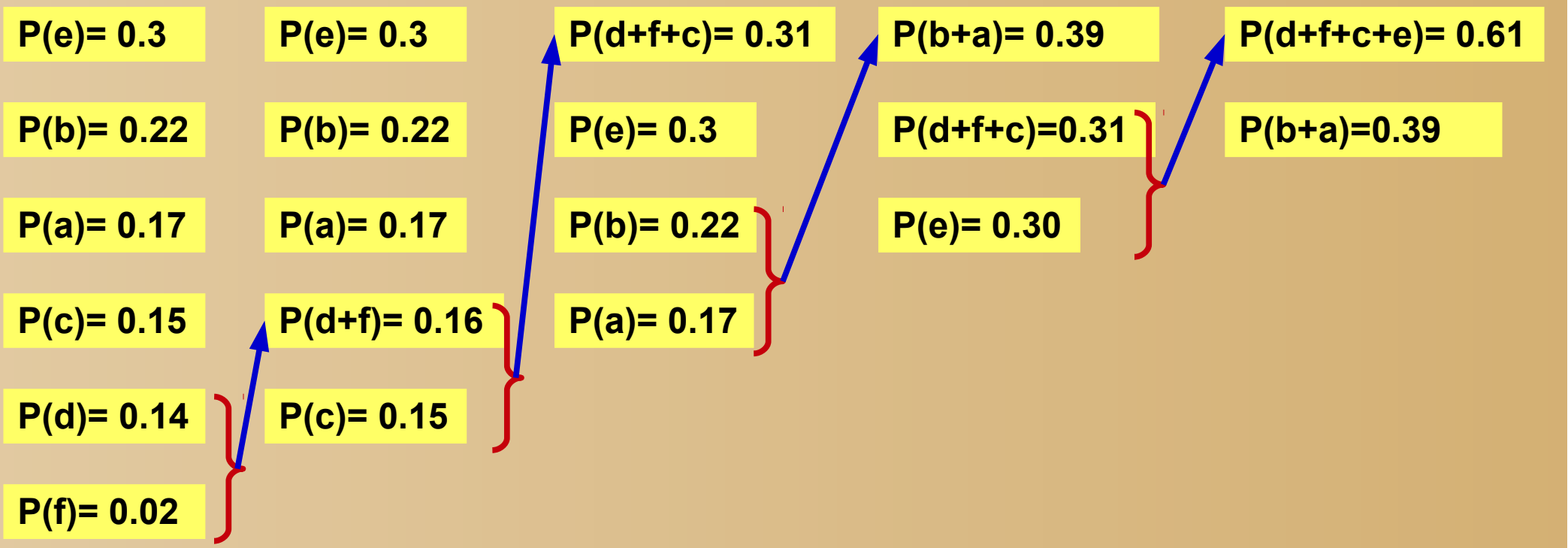


Combine Last two Symbols (with Lowest Probabilities),
Then reorder the list

Prof. Khaled Mostafa
khaledms@fci-cu.edu.eg

Huffman Coding Example

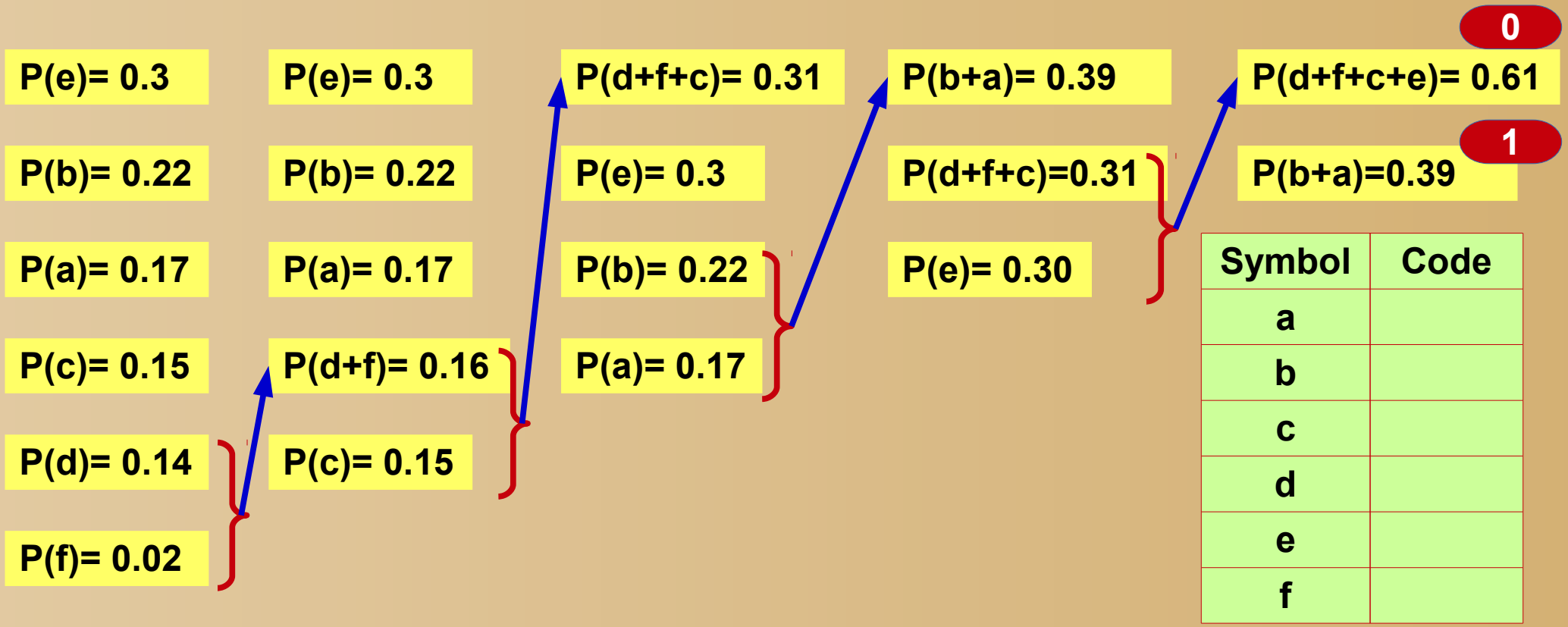
جامعة القاهرة
كلية الحاسبات
و المعلومات



Combine Last two Symbols (with Lowest Probabilities),
Then reorder the list

Prof. Khaled Mostafa
khaledms@fci-cu.edu.eg

Huffman Coding Example

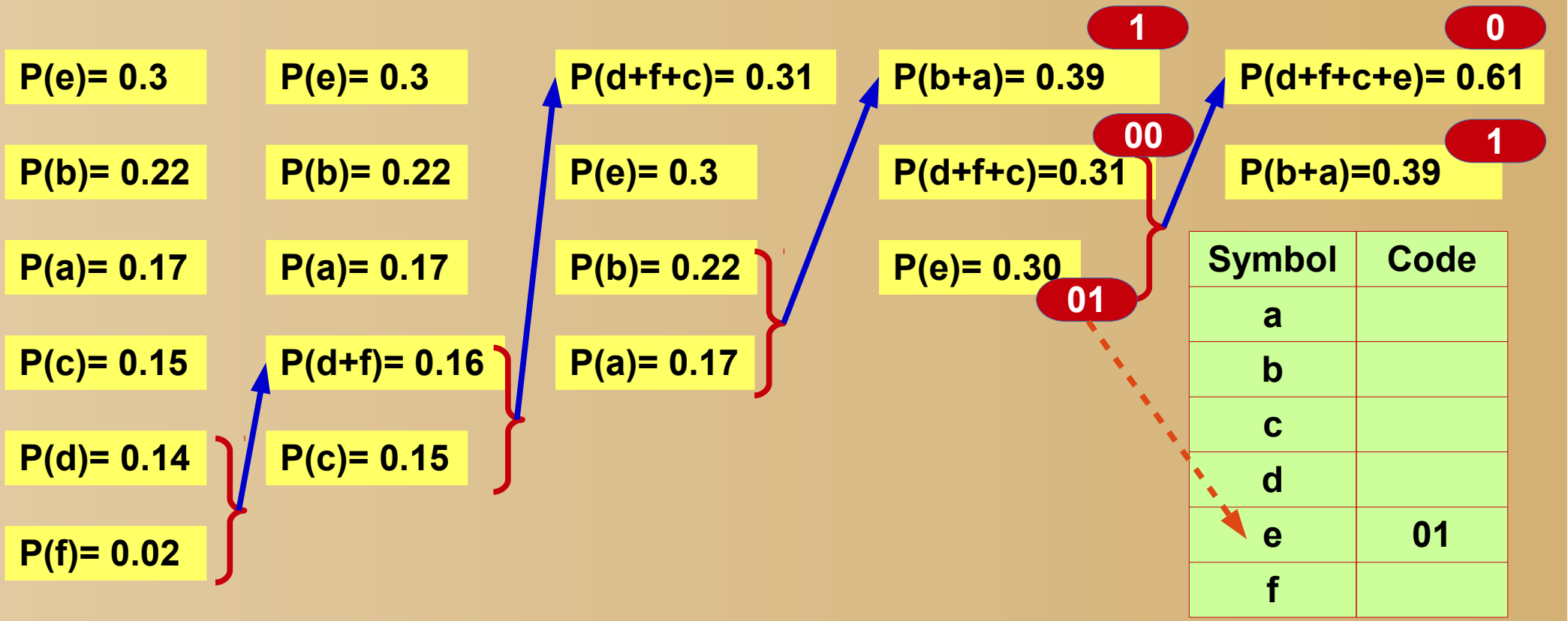


Symbol	Code
a	
b	
c	
d	
e	
f	

Assign Binary Codes to each branch, Continue

Prof. Khaled Mostafa
khaledms@fci-cu.edu.eg

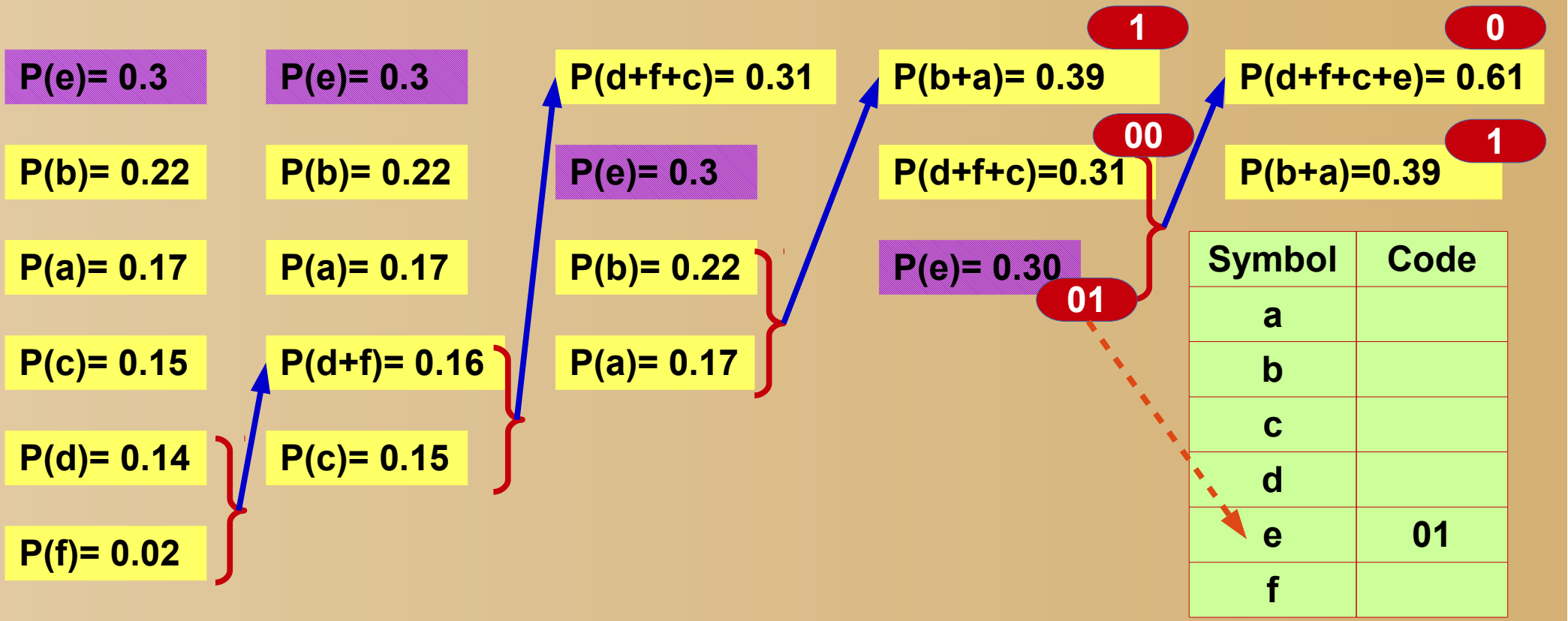
Huffman Coding Example



Assign Binary Codes to each branch, Continue

Prof. Khaled Mostafa
khaledms@fci-cu.edu.eg

Huffman Coding Example

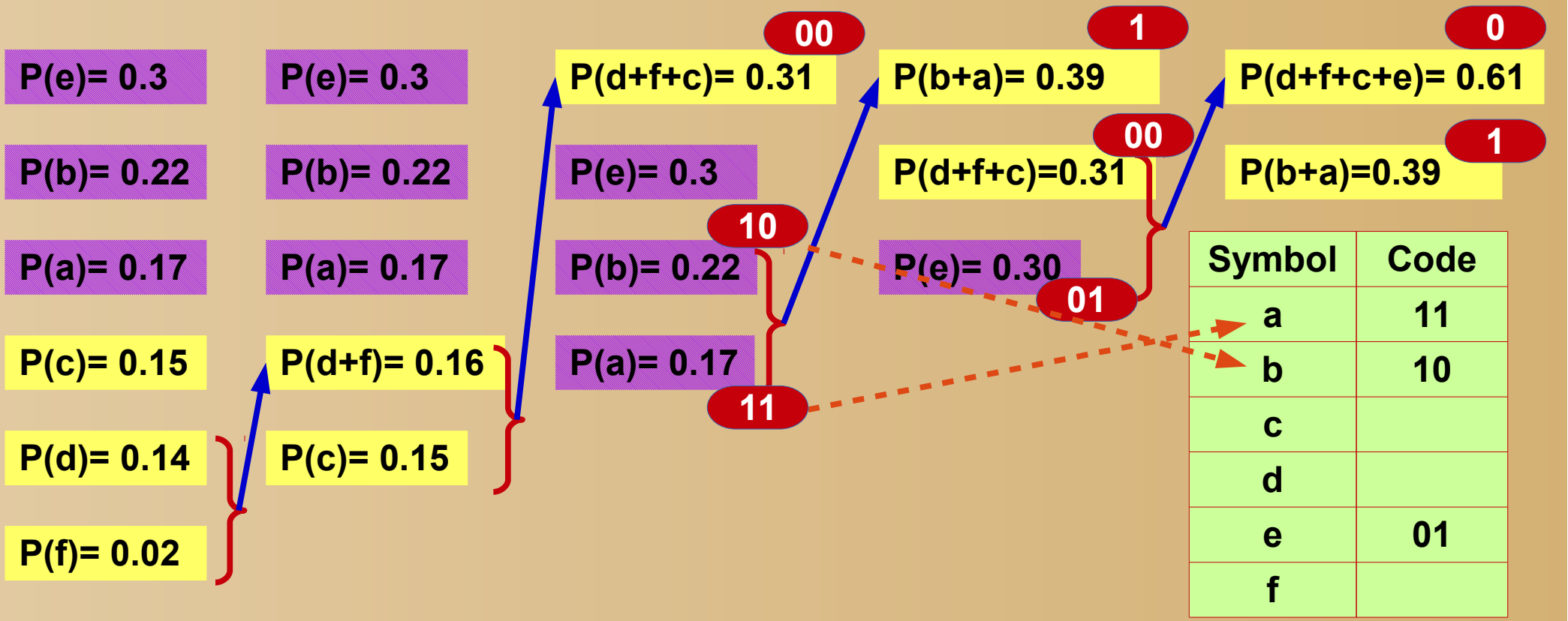


Symbol	Code
a	
b	
c	
d	
e	01
f	

Assign Binary Codes to each branch, Continue

Prof. Khaled Mostafa
khaledms@fci-cu.edu.eg

Huffman Coding Example

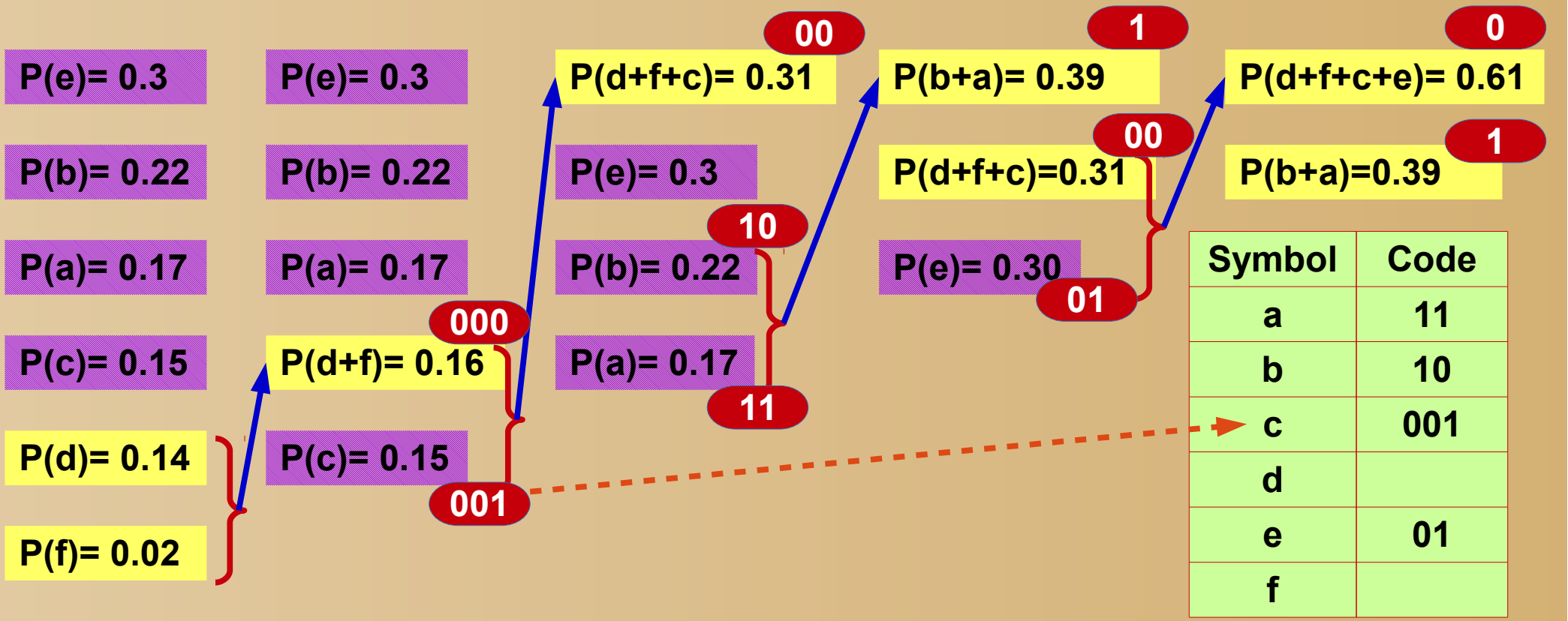


Symbol	Code
a	11
b	10
c	
d	
e	01
f	

Assign Binary Codes to each branch, Continue

Prof. Khaled Mostafa
khaledms@fci-cu.edu.eg

Huffman Coding Example

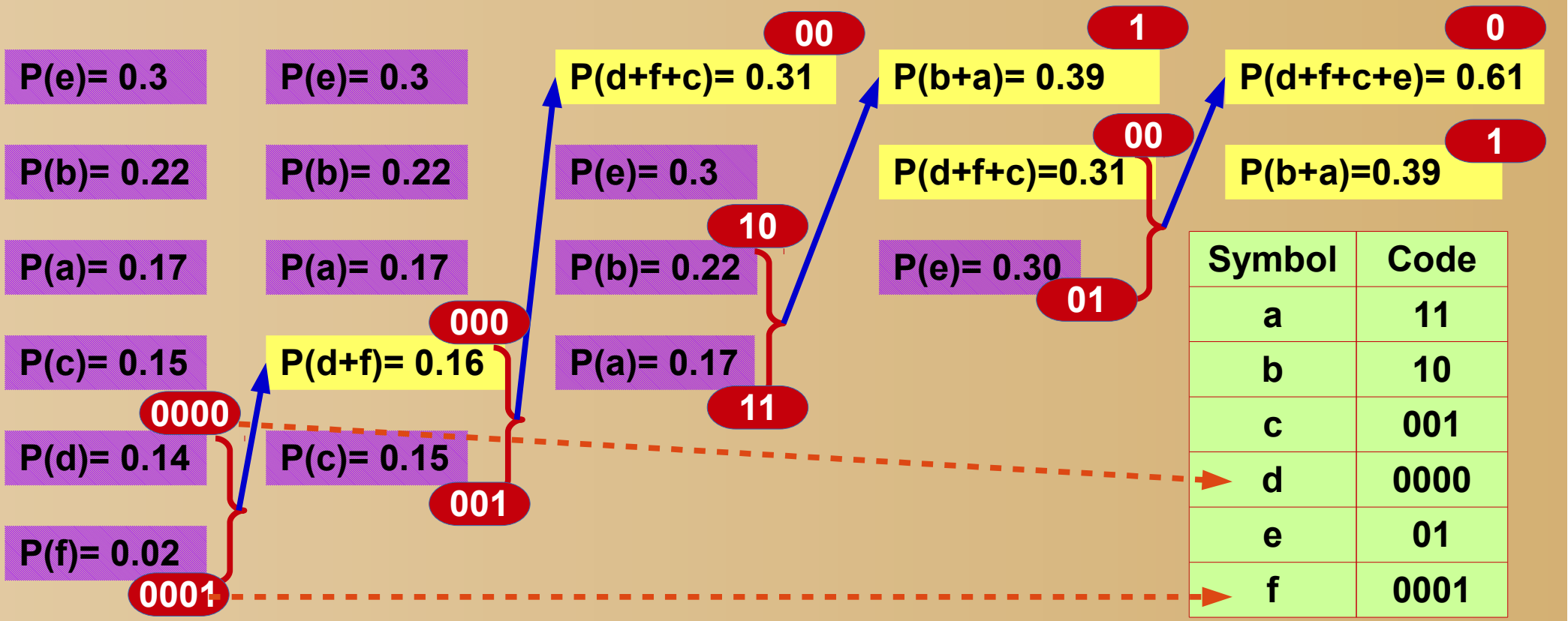


Symbol	Code
a	11
b	10
c	001
d	
e	01
f	

Assign Binary Codes to each branch, Continue

Prof. Khaled Mostafa
khaledms@fci-cu.edu.eg

Huffman Coding Example



Symbol	Code
a	11
b	10
c	001
d	0000
e	01
f	0001

Assign Binary Codes to each branch, Continue

Prof. Khaled Mostafa
khaledms@fci-cu.edu.eg

Compression Ratio

Symbol	Count	Code	Code Length	Total	Comment
a	17	11	2	34	
b	22	10	2	44	
c	15	001	3	45	
d	14	0000	4	56	
e	30	01	2	60	
f	2	0001	4	8	

Compressed Total = 247 Bits

Uncompressed size=100 Symbol * 3 bits/symbol = 300 Bits

Entropy =2.35 bits/Symbol (for 100 Symbols H=235 bits

Prof. Khaled Mostafa
khaledms@fci-cu.edu.eg

Example 2: Huffman Code Construction

In the following Table

Given ***Data Symbols (Character)*** and

Corresponding ***Number of Occurrence (Frequency)***

Use Standard Huffman Coding to generate

VLC for each Symbol

Char	Freq
E	125
T	93
A	80
O	76
I	72
N	71
S	65
R	61
H	55
L	41
D	40
C	31
U	27

Prof. Khaled Mostafa
khaledms@fci-cu.edu.eg

Example 2: Huffman Code Construction

Prof. Khaled Mostafa
khaledms@fci-cu.edu.eg

Example2: Huffman Code Construction

Prof. Khaled Mostafa
khaledms@fci-cu.edu.eg

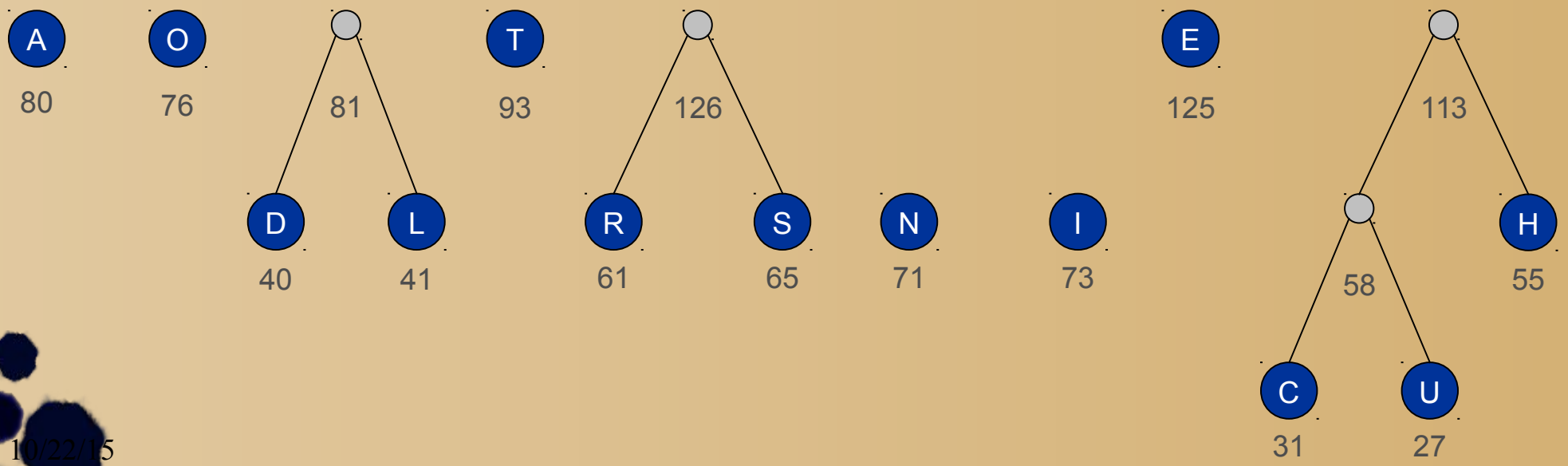
Example2: Huffman Code Construction

Prof. Khaled Mostafa
khaledms@fci-cu.edu.eg

Example2: Huffman Code Construction

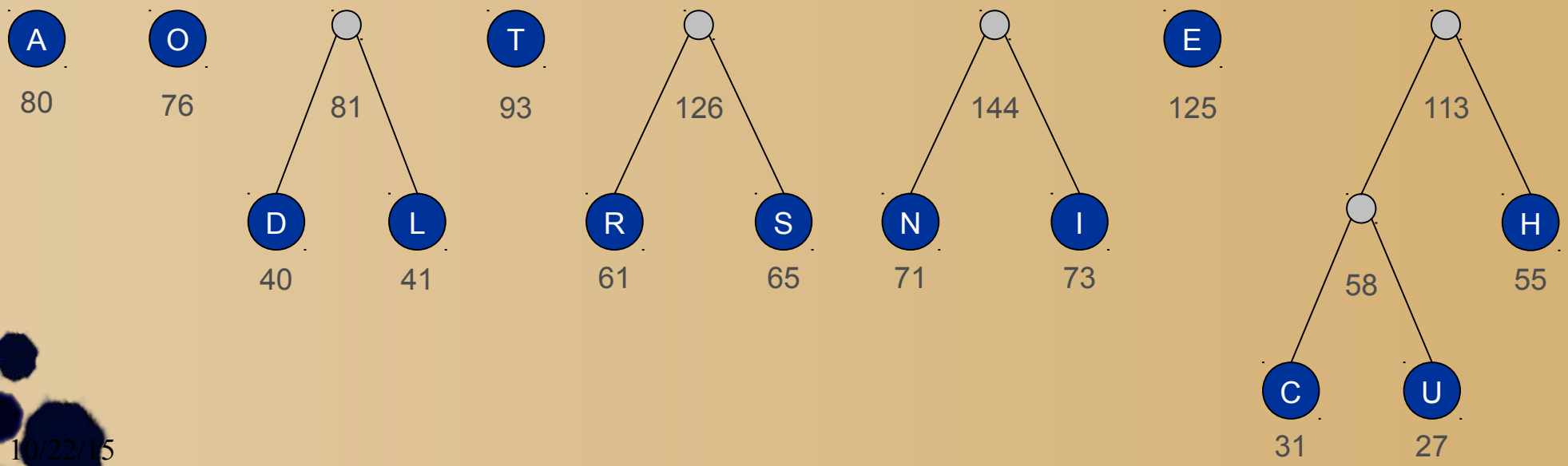
Prof. Khaled Mostafa
khaledms@fci-cu.edu.eg

Example2: Huffman Code Construction



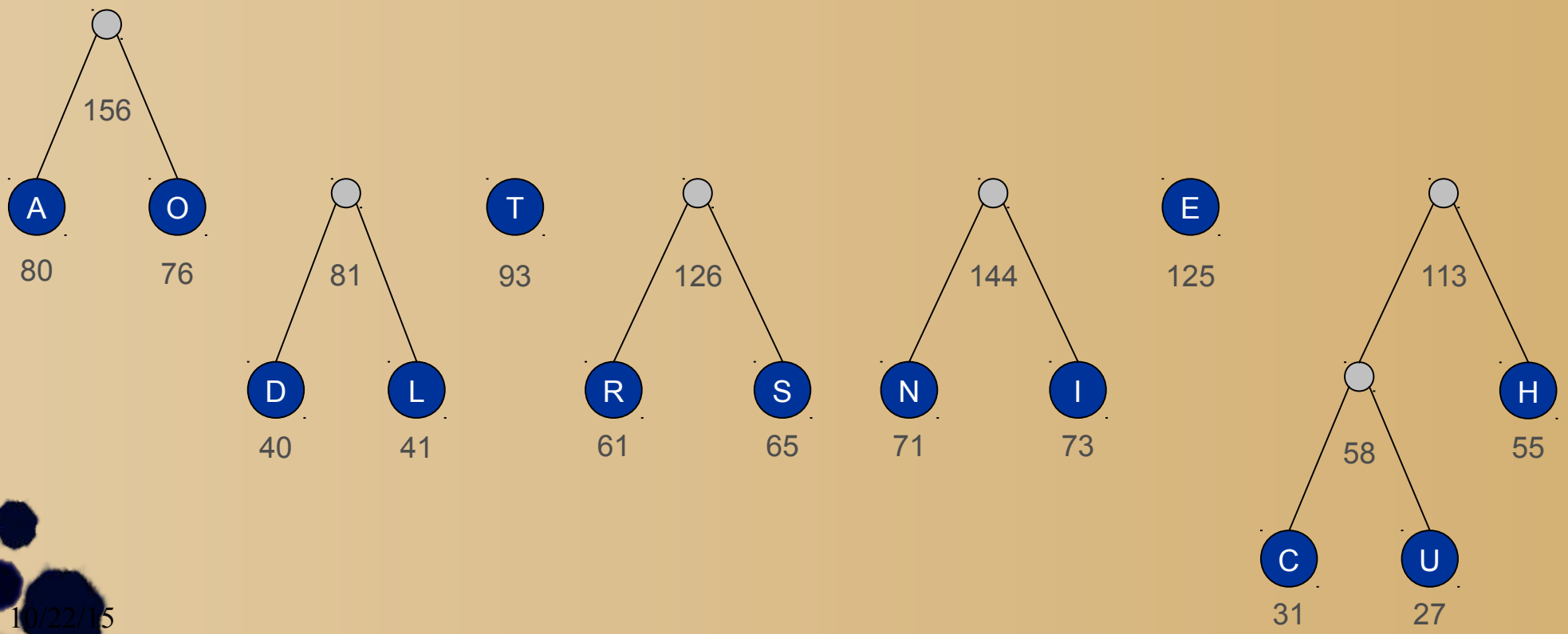
Prof. Khaled Mostafa
khaledms@fci-cu.edu.eg

Example2: Huffman Code Construction



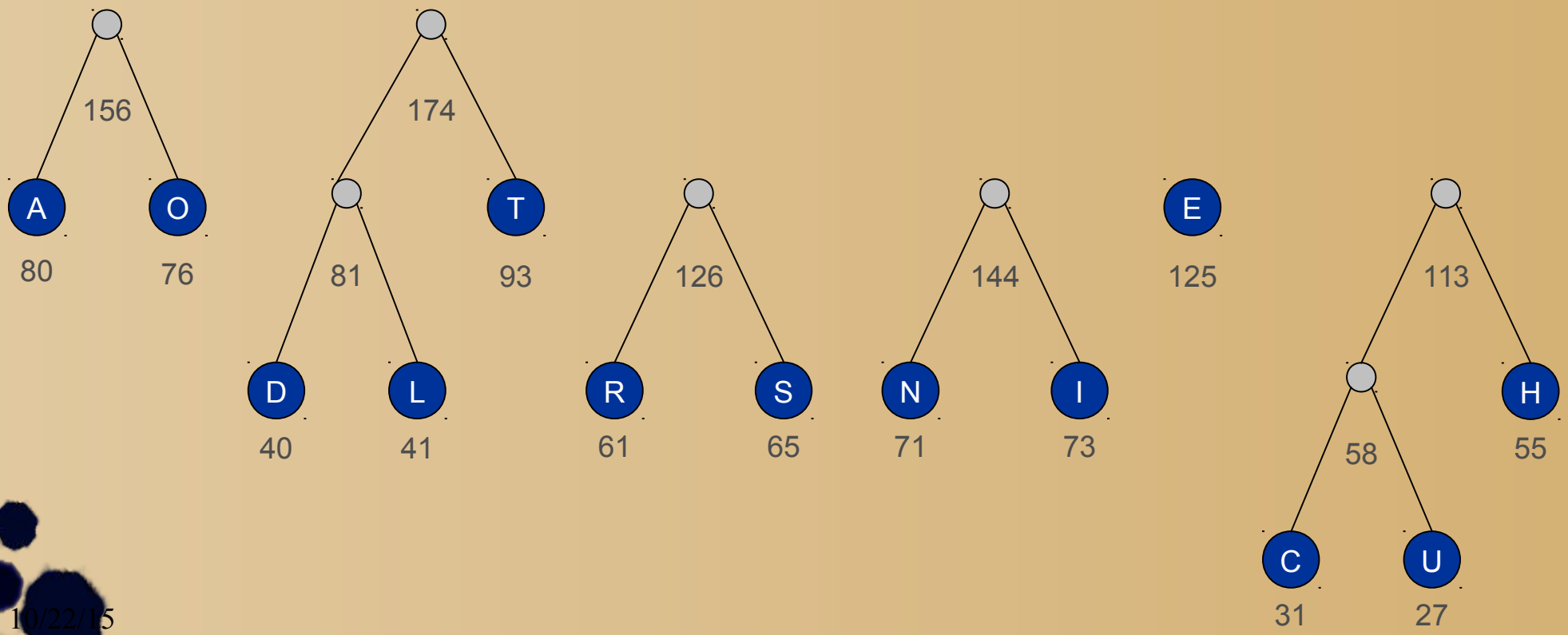
Prof. Khaled Mostafa
khaledms@fci-cu.edu.eg

Example2: Huffman Code Construction



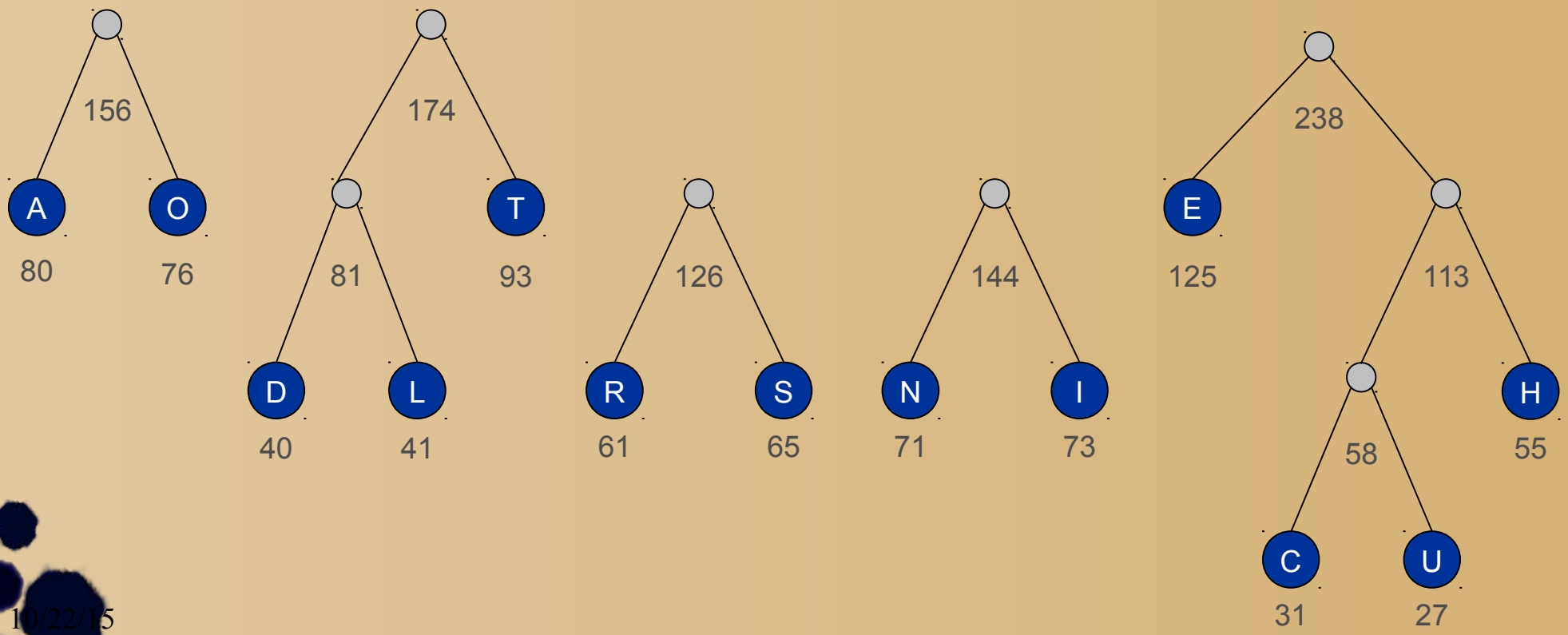
Prof. Khaled Mostafa
khaledms@fci-cu.edu.eg

Example2: Huffman Code Construction



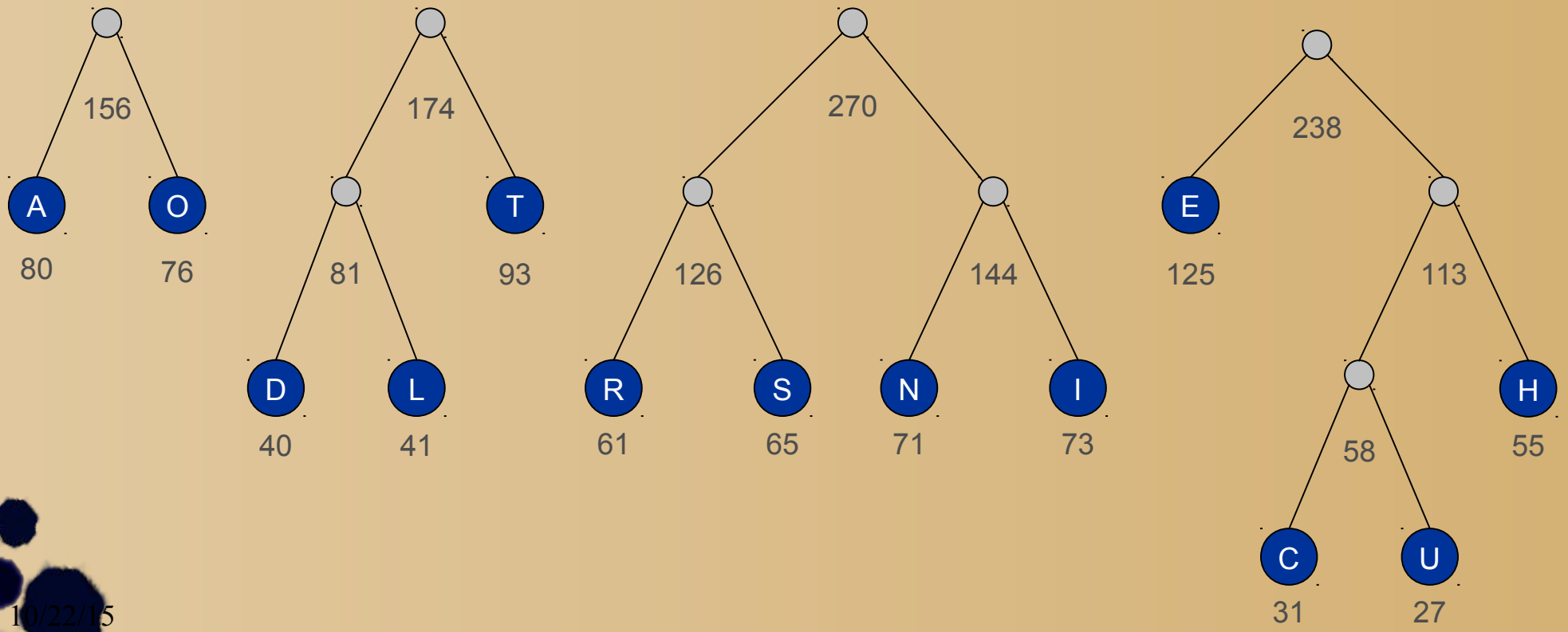
Prof. Khaled Mostafa
khaledms@fci-cu.edu.eg

Example2: Huffman Code Construction



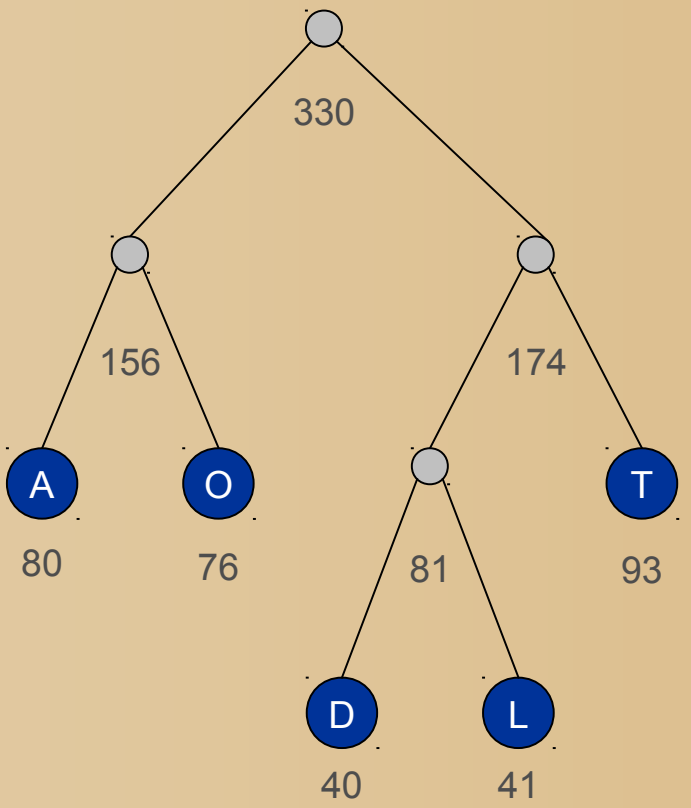
Prof. Khaled Mostafa
khaledms@fci-cu.edu.eg

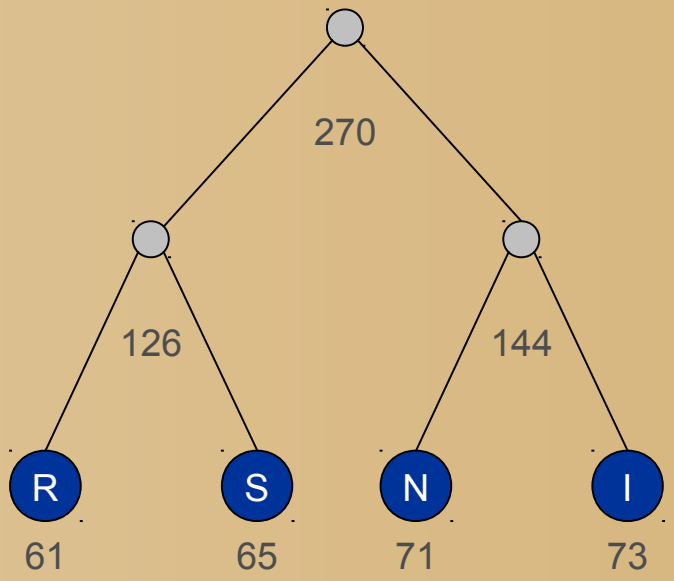
Example2: Huffman Code Construction

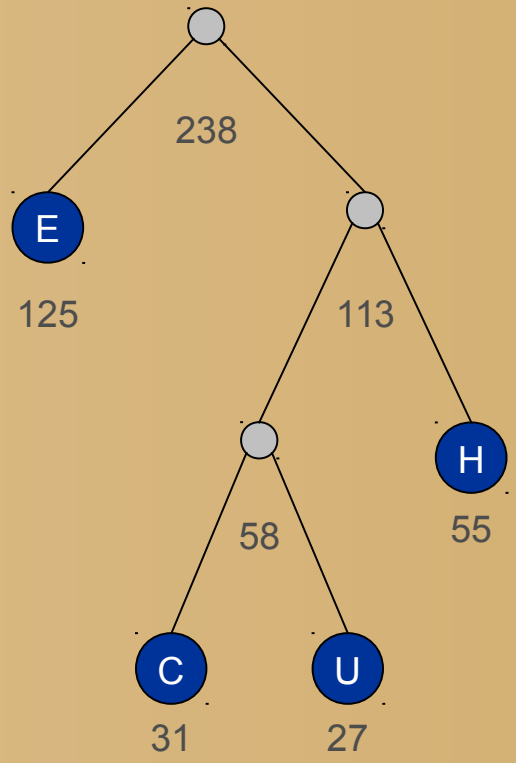


Prof. Khaled Mostafa
khaledms@fci-cu.edu.eg

Example2: Huffman Code Construction

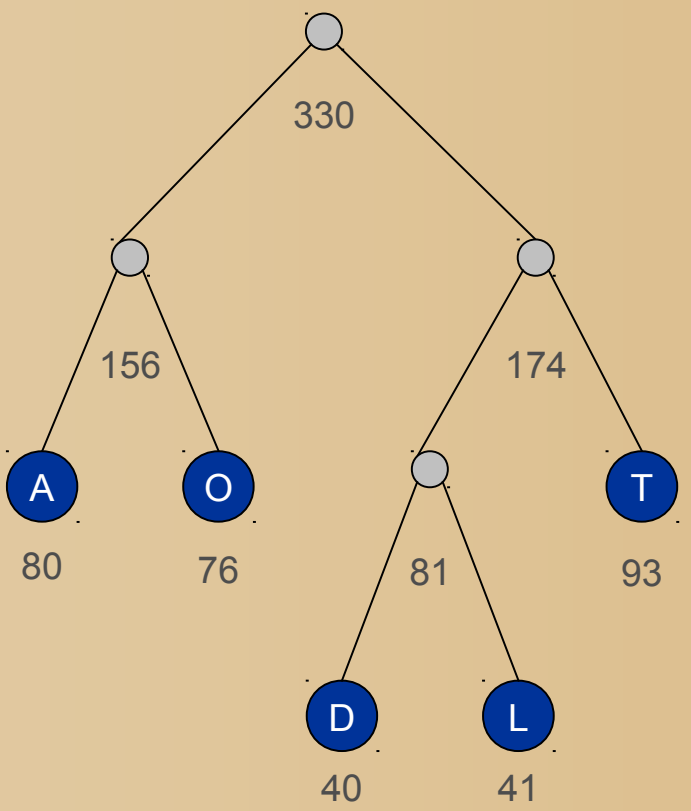


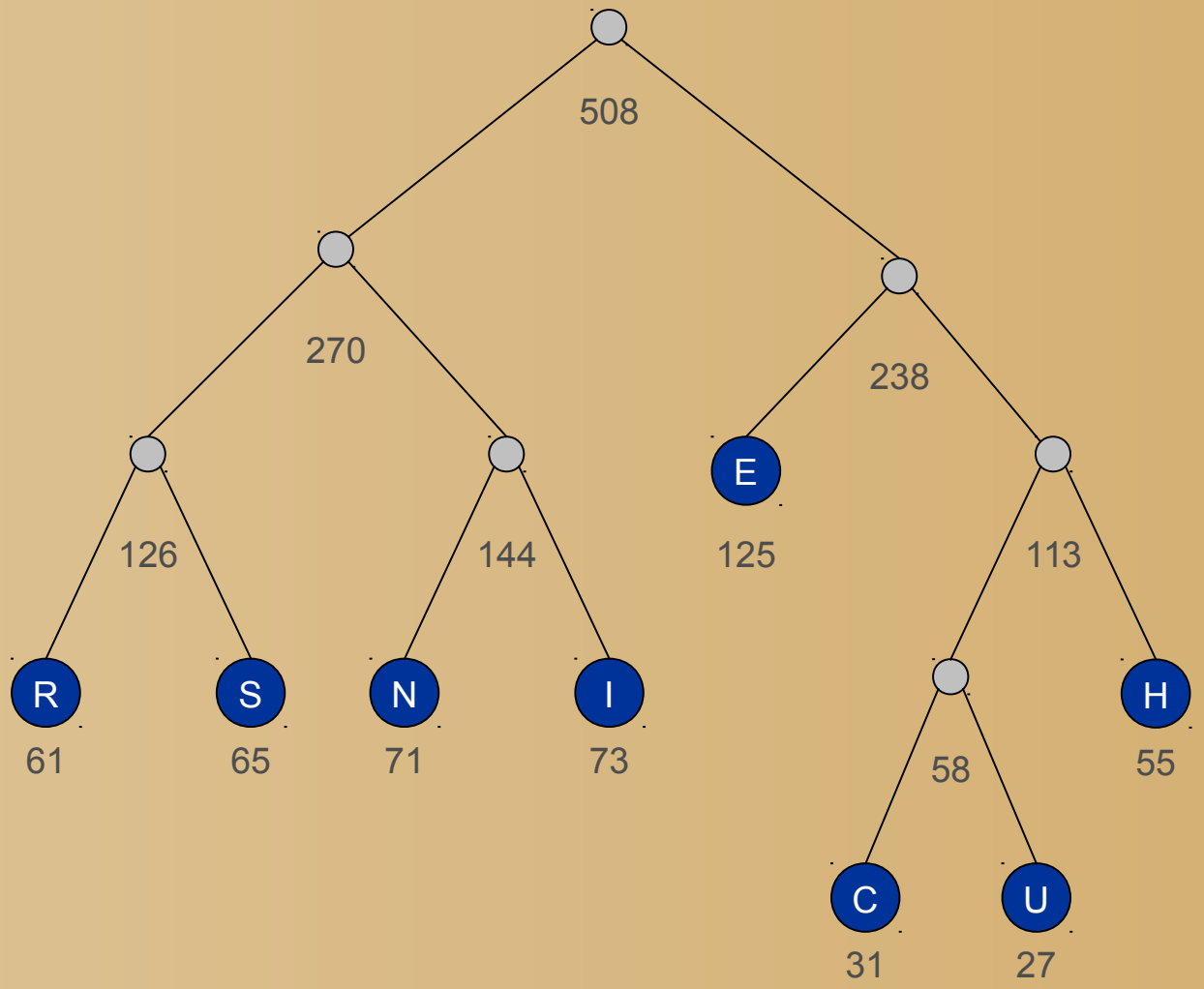




Prof. Khaled Mostafa
khaledms@fci-cu.edu.eg

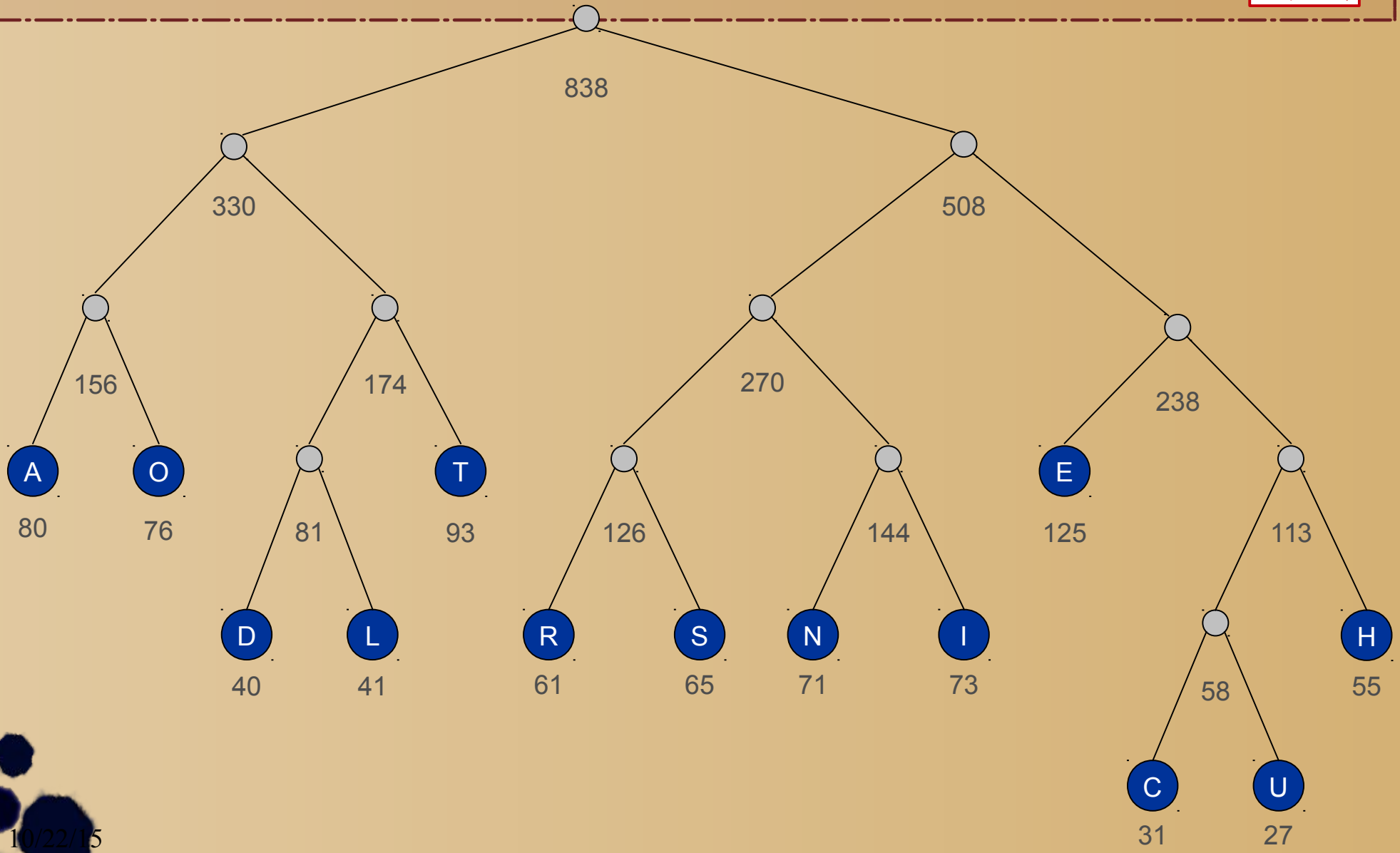
Example2: Huffman Code Construction





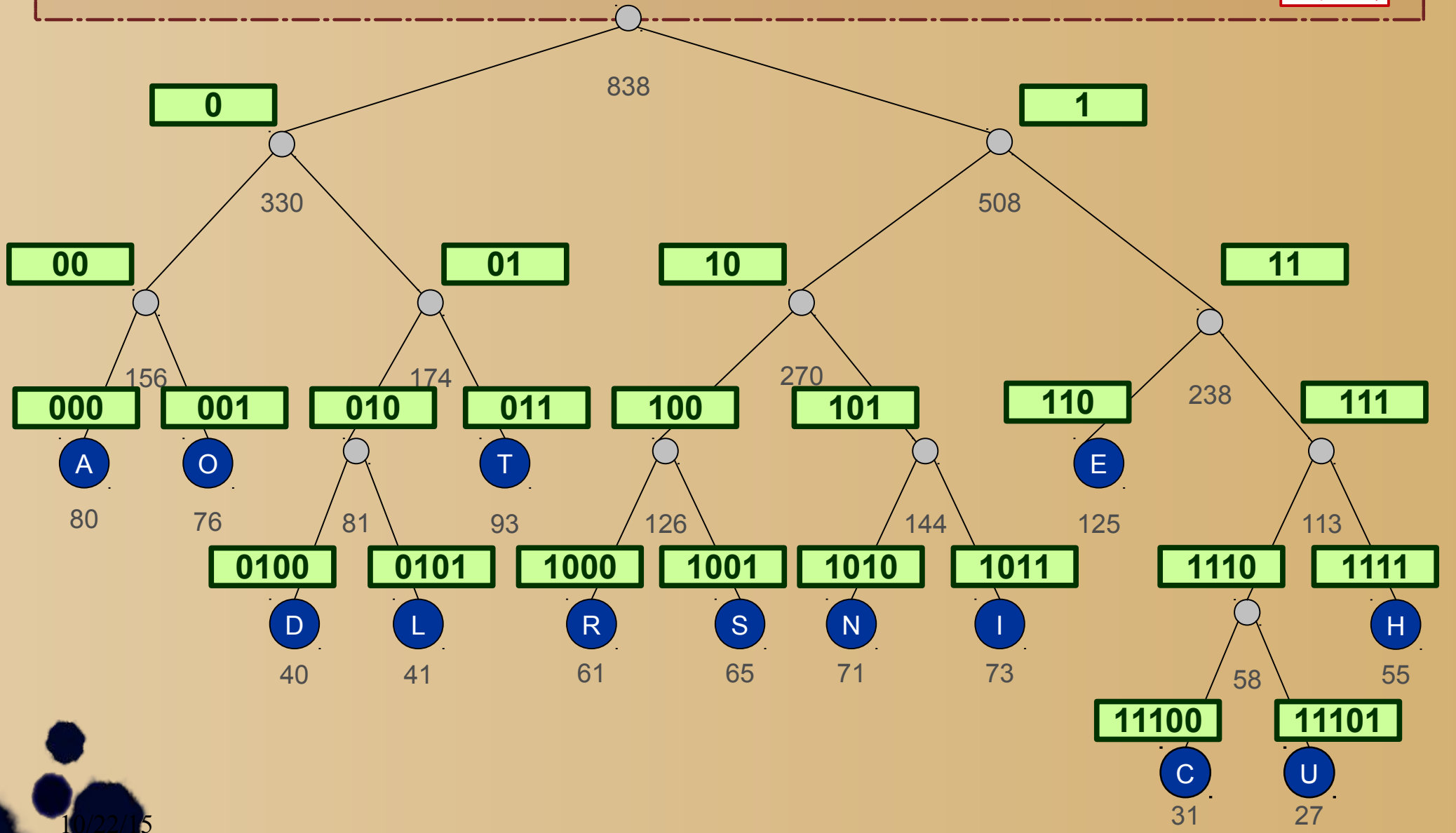
Prof. Khaled Mostafa
khaledms@fci-cu.edu.eg

Example2: Huffman Code Construction



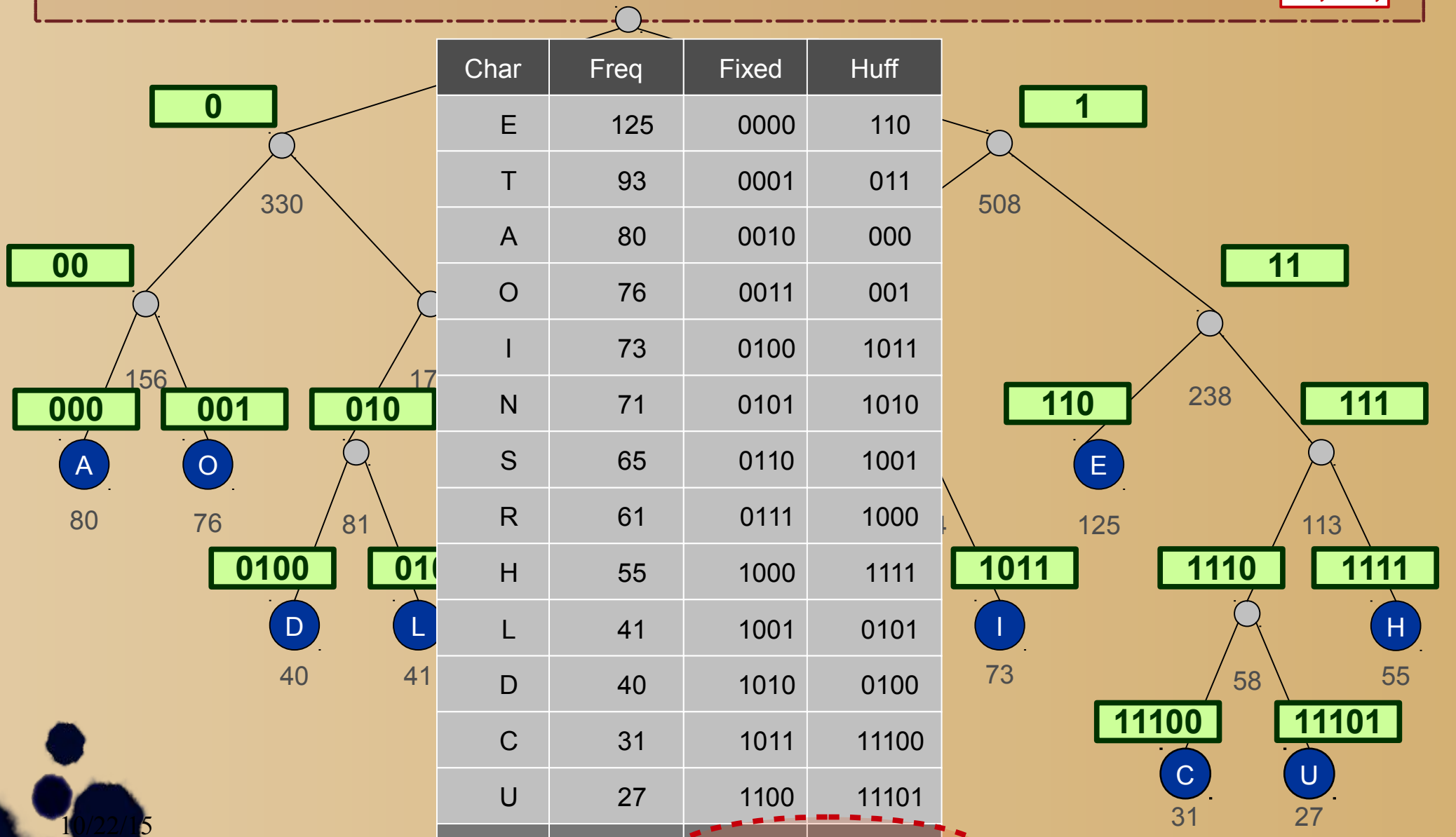
Prof. Khaled Mostafa
khaledms@fci-cu.edu.eg

Example Huffman Code Construction



Prof. Khaled Mostafa
khaledms@fci-cu.edu.eg

Example Huffman Code Construction



Char	Freq	Fixed	Huff
E	125	0000	110
T	93	0001	011
A	80	0010	000
O	76	0011	001
I	73	0100	1011
N	71	0101	1010
S	65	0110	1001
R	61	0111	1000
H	55	1000	1111
L	41	1001	0101
D	40	1010	0100
C	31	1011	11100
U	27	1100	11101
Total	838	4.00	3.62

Bits/Symbol

Prof. Khaled Mostafa
khaledms@fci-cu.edu.eg

جامعة القاهرة
كلية الحاسبات
و المعلومات

Modified Huffman Coding Algorithm

Prof. Khaled Mostafa
khaledms@fci-cu.edu.eg

Modified Huffman Coding Algorithm

- Same steps and concept as Huffman Coding Algorithm
- In order to Minimize Huffman Table size and Codes lengths, set **Minimum Limit of Symbol Probabilities** (e.g. 0.05)
- Symbols with Probabilities <= the Limit will be grouped in one group called "**Others**"
- All Symbols will be coded using Corresponding Huffman codes, Symbols in "Other" group will be coded using both **"Others" Huffman code** + **Original** Symbol Code.

Prof. Khaled Mostafa
khaledms@fci-cu.edu.eg

Huffman Coding Example

a b c a z d a f c q d a d c u a b a p d

Count (a) = 6	P(a)= 0.3
Count (b) = 2	P(b)= 0.1
Count (c) = 3	P(c)= 0.15
Count (d) = 4	P(d)= 0.2
Count (f) = 1	P(f)= 0.05
Count (z) = 1	P(z)= 0.05
Count (q) = 1	P(q)= 0.05
Count (p) = 1	P(p)= 0.05
Count (u) = 1	P(u)= 0.05

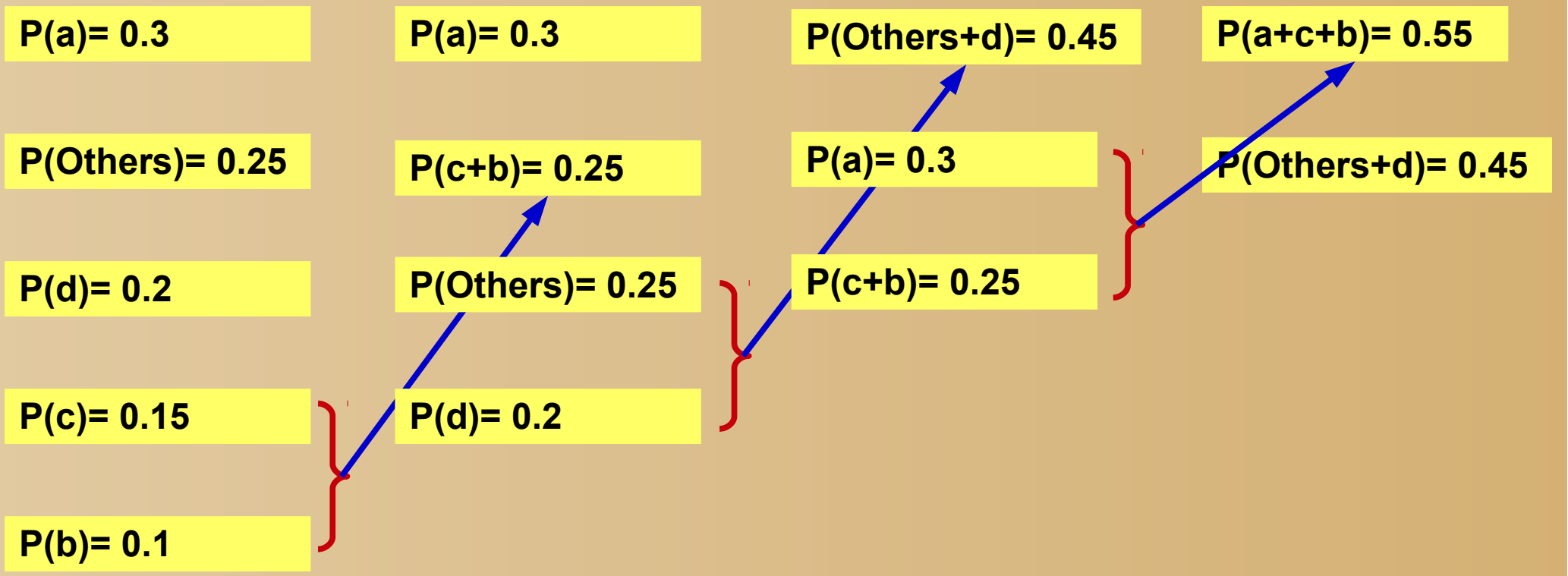
P(a)= 0.3
P(b)= 0.1
P(c)= 0.15
P(d)= 0.2
P(Others)= 0.25

Symbol	Original Code
a	0000
b	0001
c	0010
d	0011
f	0100
p	0101
q	0110
u	0111
z	1000

P
khateams@fci-cu.eau.eg

Modified Huffman Coding Example

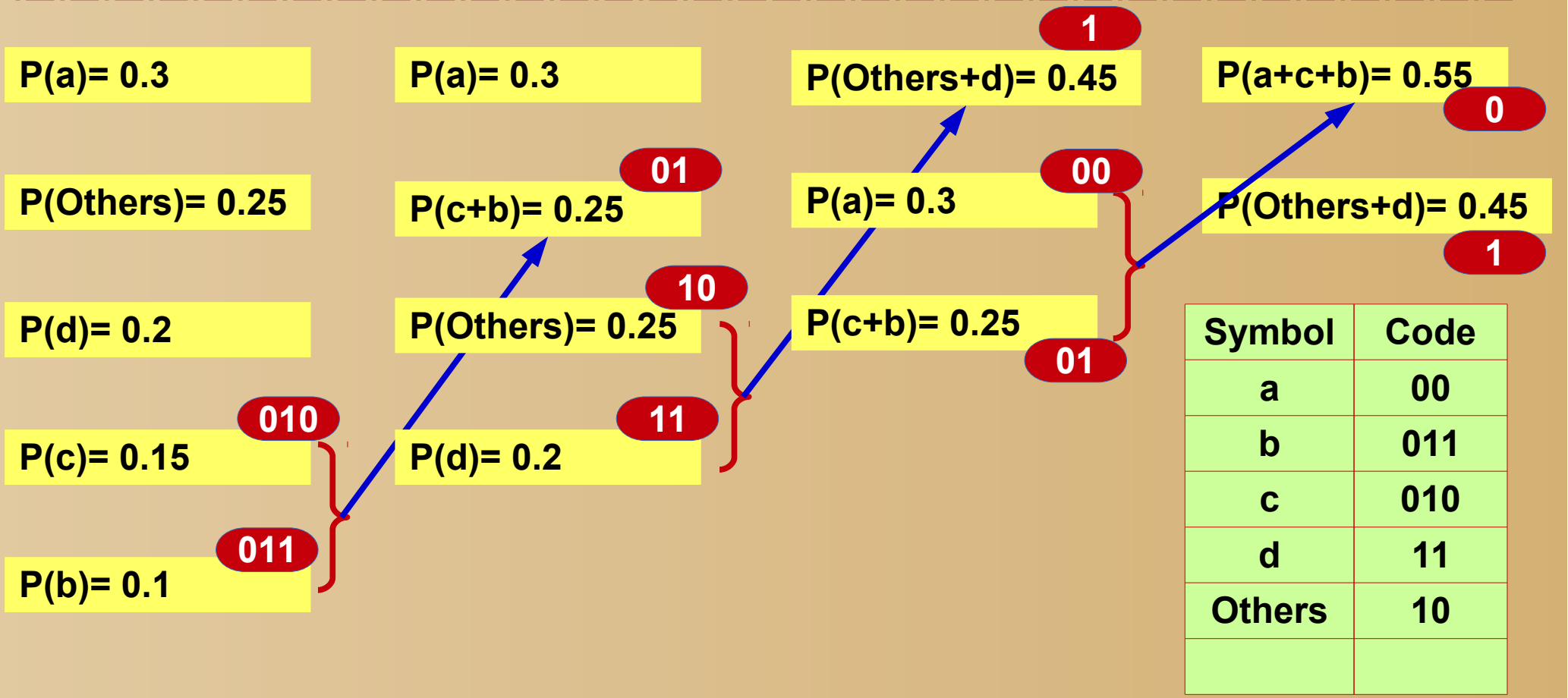
جامعة القاهرة
كلية الحاسبات
و المعلومات



Combine Last two Symbols (with Lowest Probabilities), Then reorder the list

Prof. Khaled Mostafa
khaledms@fci-cu.edu.eg

Modified Huffman Coding Example

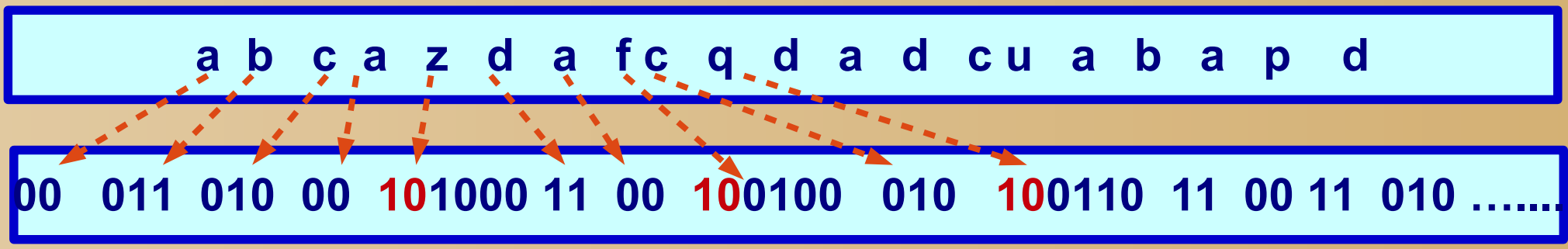


Symbol	Code
a	00
b	011
c	010
d	11
Others	10

Assign Binary Codes to each branch, Continue

Prof. Khaled Mostafa
khaledms@fci-cu.edu.eg

Modified Huffman Coding Example



A	b	c	d	Others
6*2 +	2*3 +	3*3 +	4*2 +	5 (2+4) =

12 + 6 + 9 + 8 + 30 = 65 bits

Symbol	Code
a	00
b	011
c	010
d	11
Others	10

Symbol	Original Code
a	0000
b	0001
c	0010
d	0011
f	0100
p	0101
q	0110
u	0111
z	1000

Prof. Khaled Mostafa
khaledms@fci-cu.edu.eg

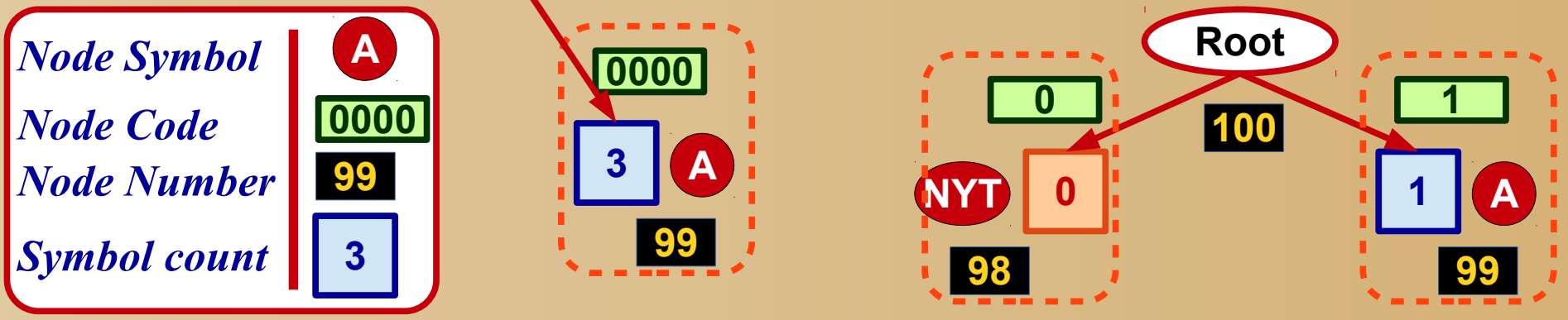
جامعة القاهرة
كلية الحاسبات
و المعلومات

Adaptive Huffman Coding Algorithm

(Unknown Symbols Probabilities)

Prof. Khaled Mostafa
khaledms@fci-cu.edu.eg

Single Node Structure



Node Symbol: *Each Node is associated with ONE Symbol. Also, Max ONE Node for each symbol*

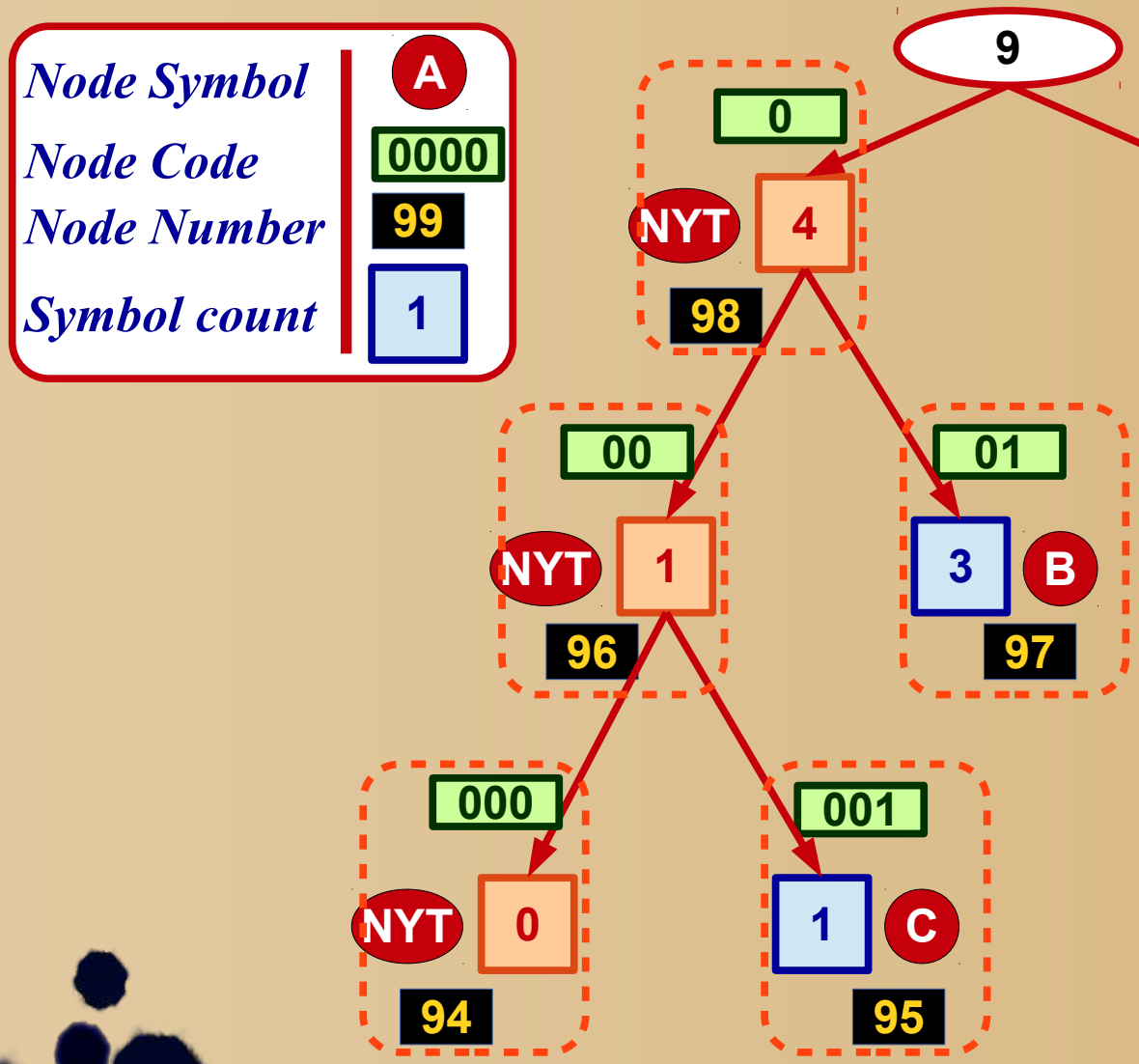
Node Count: *is the Number of Occurrence of Node Symbol (e.g. Number of Occurrence of "A")*

Node Code: *Code Nodes as Binary Tree. Use same Enumeration for Compression and Decompression (e.g. Always "1" for Right branch and "0" for left branch)*

Node Number: *Enumerate Nodes in descending order (starting from Root, 1st level Right branch, 1st level Left branch , then for 2nd level Right branch ,...etc)*

Prof. Dr. Khaled Mostafa El-Sayed
khaledms@fci-cu.edu.eg

Adaptive Huffman Main Concept



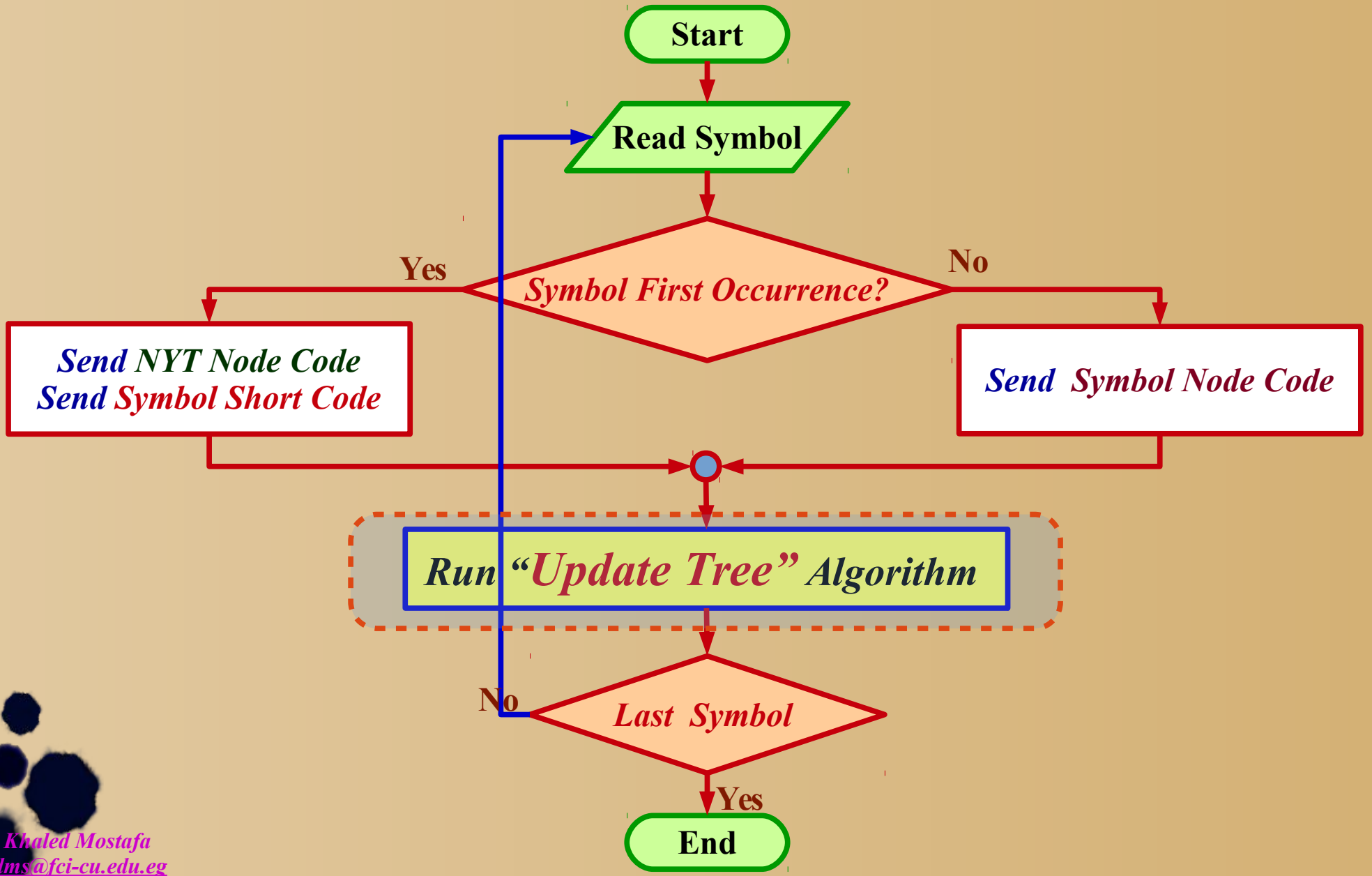
Symbol	Short Code
A	00
B	01
C	10

- *Symbols at higher Tree level (Bigger Node Number) have shorter Code*
- *Periodically Swap Symbol Nodes in order to keep Symbols with higher Occurrence at higher level in the tree*
- *For first symbol Occurrence, use Symbol Short Code (only Once)*

Prof. Khaled Mostafa
khaledms@fci-cu.edu.eg

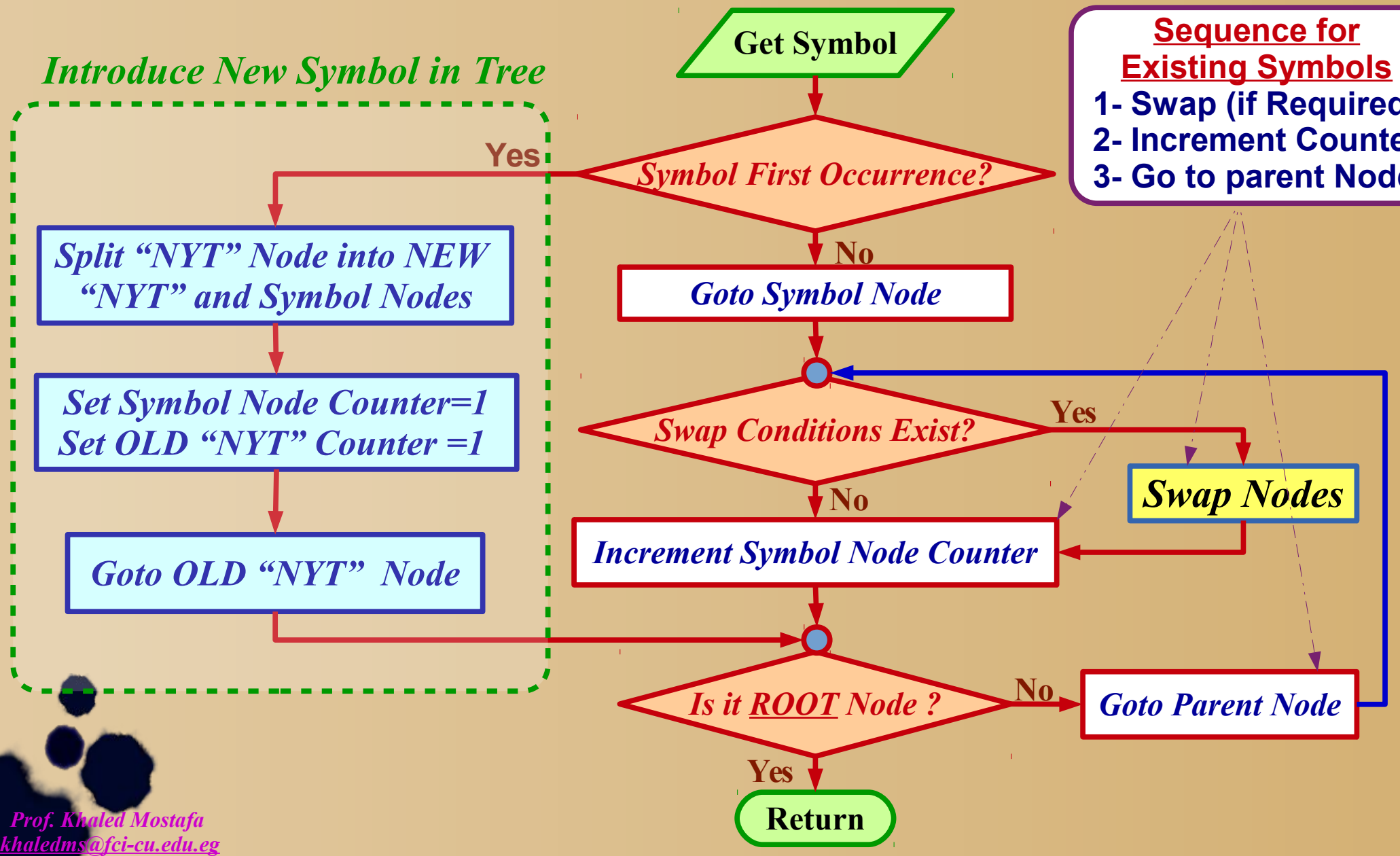
Adaptive Huffman Algorithm

جامعة القاهرة
كلية الحاسبات
و المعلومات



Prof. Khaled Mostafa
khaledms@fci-cu.edu.eg

"Update Tree" Algorithm



Prof. Khaled Mostafa
khaledms@fci-cu.edu.eg

Swap Conditions

If you are a Node of symbol "X"

You can Swap Node of Symbol "X" with Node of Symbol "Y" if

1- **Node Number** of Symbol "Y" is higher than Node Number of Symbol "X"

2- **Node counter** of Node "X" is higher than or equal to Node Counter of Node "Y"

3- **No Swap with Parent Node**

4- if "X" can Swap with more than one Node, **Swap with Node with higher Node Number**

Prof. Khaled Mostafa
khaledms@fci-cu.edu.eg

Swap Conditions

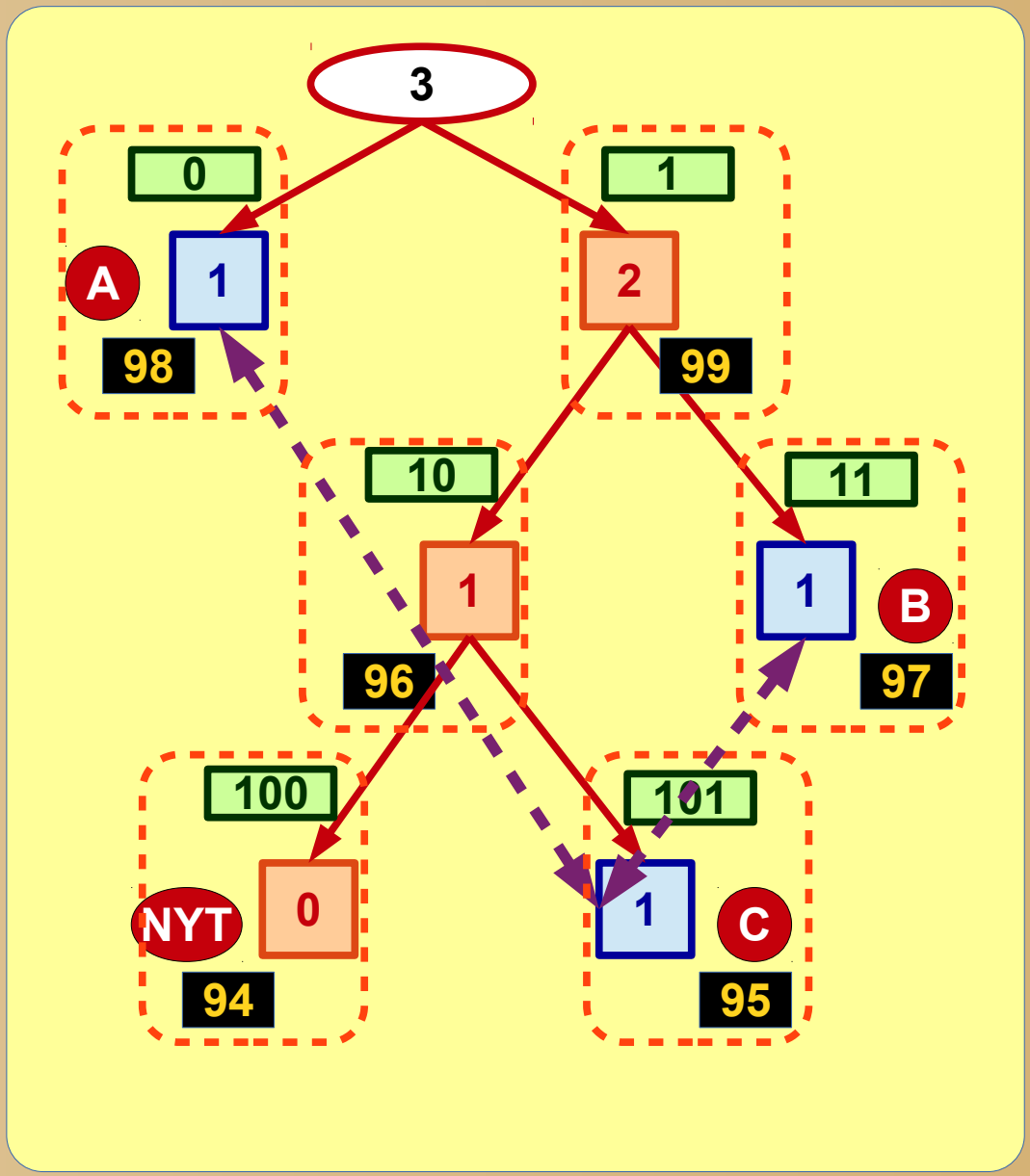
جامعة القاهرة
كلية الحاسبات
و المعلومات

Example of Allowed Swaps:

Swap Node 95 with Node 98
Swap Node 95 with Node 97
(if both are applicable, swap with 98)

Example of NOT Allowed Swaps:

Node 95 with Node 96 **{Parent}**
Node 98 with Node 99 **{Counter}**

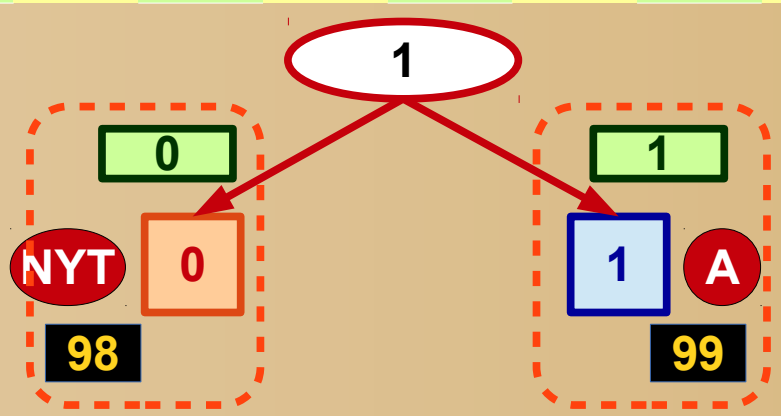


Prof. Khaled Mostafa
khaledms@fci-cu.edu.eg

Example: Adaptive Huffman Compression

Symbol	"A"														
Code	00														

Symbol	Short Code
A	00
B	01
C	10



ABCCCAAAA

Code

Symbol Occurs for first time:

Code ="A" Short Code [00]

Update Tree

Go To Node: Root

Split: Yes

Inc Counter: Node 99,Root

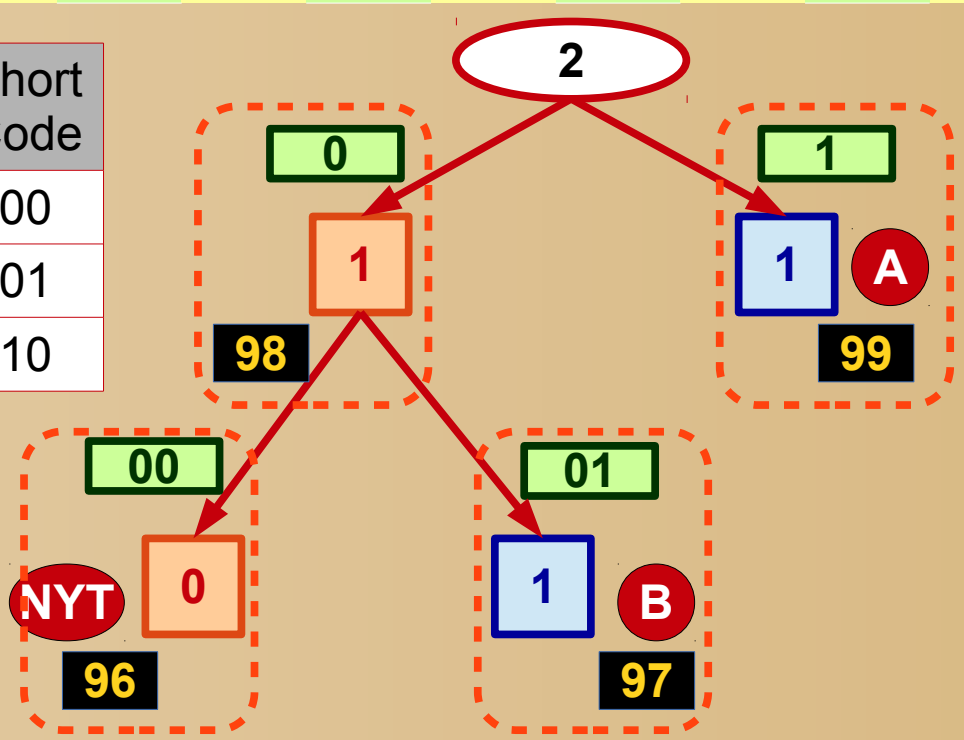
Goto Parent: Root

Prof. Khaled Mostafa
khaledms@fci-cu.edu.eg

Example: Adaptive Huffman Compression

Symbol	"A"	NYT	"B"												
Code	00	0	01												

Symbol	Short Code
A	00
B	01
C	10



ABCCCAAAA

Code

Symbol Occurs for first time:

Code = NYT [0],

"B"Short Code [01]

Update Tree

Go To Node: NYT [98]

Split: Yes

Inc Counter: Node 97, 98

Goto Parent: Root,

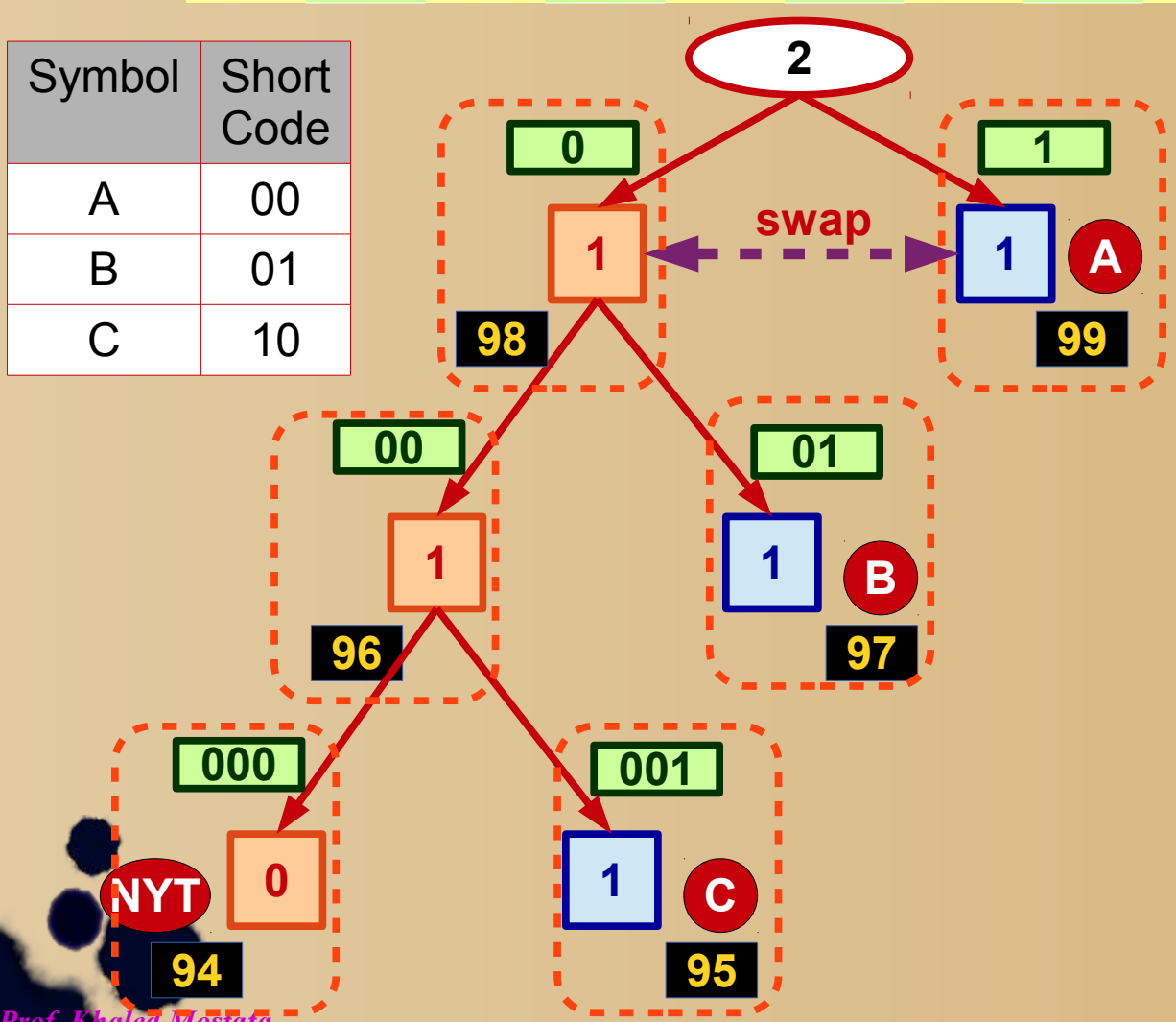
Inc {1+}

Prof. Khaled Mostafa
khaledms@fci-cu.edu.eg

Example: Adaptive Huffman Compression

Symbol	"A"	NYT	"B"	NYT	"C"										
Code	00	0	01	00	10										

Symbol	Short Code
A	00
B	01
C	10



ABCCCAAAA

Code

Symbol Occurs for first time:

Code = NYT [00],

"C" Short Code [10]

Update Tree

Go To Node: NYT [96]

Split: Yes

Inc Counter: Node 95, 96

Goto Parent: 98

Need Swap: Yes [98], [99]

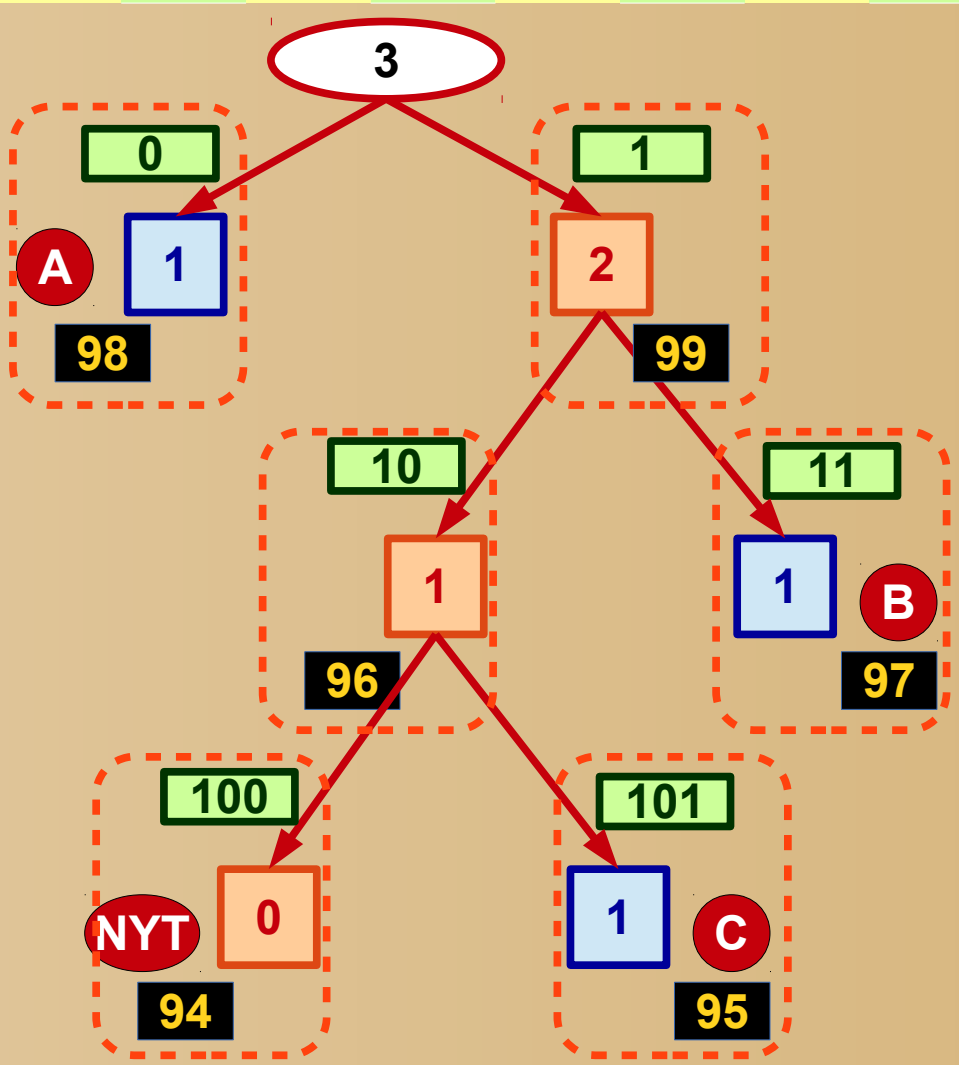
Prof. Khaled Mostafa
khaledms@fci-cu.edu.eg

Example: Adaptive Huffman Compression

جامعة القاهرة
كلية الحاسبات و المعلومات

Symbol	"A"	NYT	"B"	NYT	"C"										
Code	00	0	01	00	10										

Symbol	Short Code
A	00
B	01
C	10



ABCCCAAAA

Code

Update Tree

Swap: Done [98], [99]
Inc Counter: Node 99 [1+]
Goto Parent: Root,
Inc {2+}

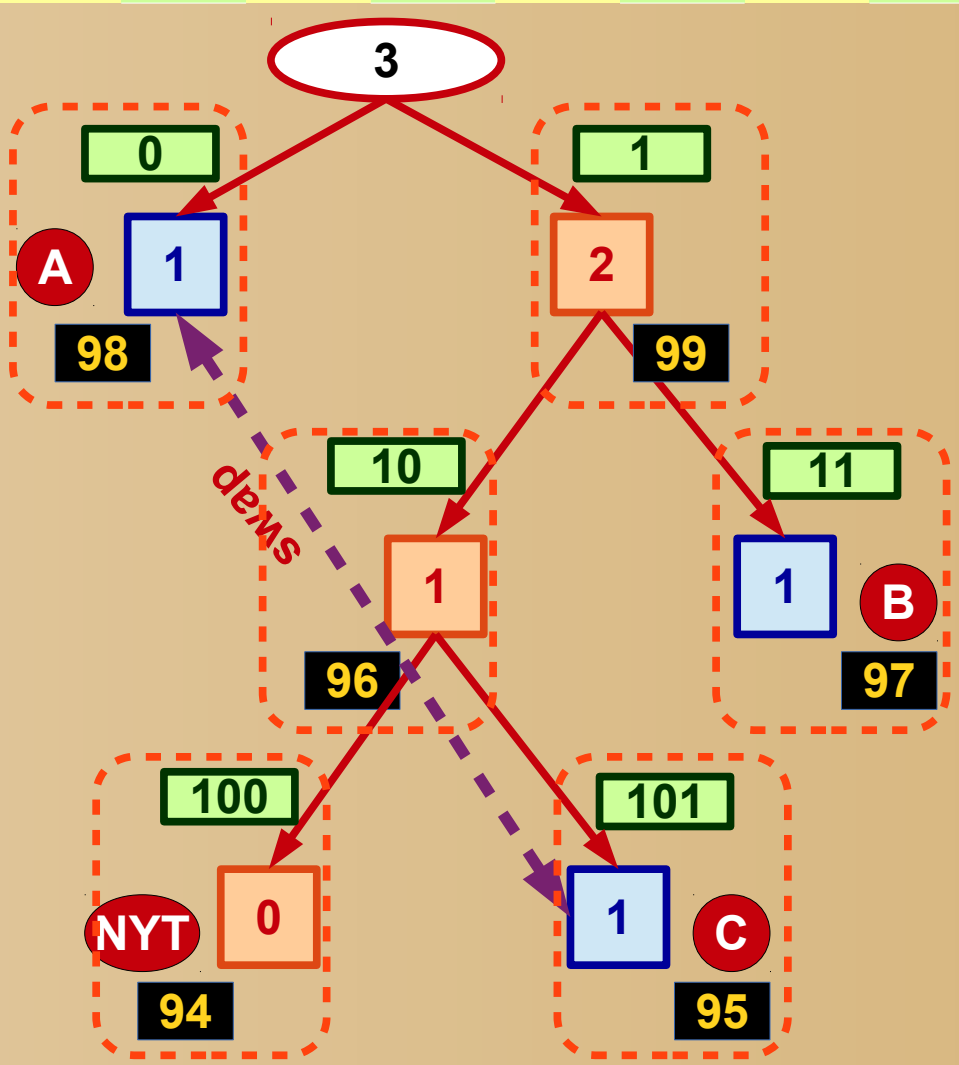
Prof. Khaled Mostafa
khaledms@fci-cu.edu.eg

Example: Adaptive Huffman Compression

جامعة القاهرة
كلية الحاسبات
و المعلومات

Symbol	"A"	NYT	"B"	NYT	"C"	"C"									
Code	00	0	01	00	10	101									

Symbol	Short Code
A	00
B	01
C	10



ABCCCAAAA

Code

Symbol Occurred before:
Node: [95]
Code =[101]

Update Tree

Go To Node: 95
Need Swap: YES [95], [98]

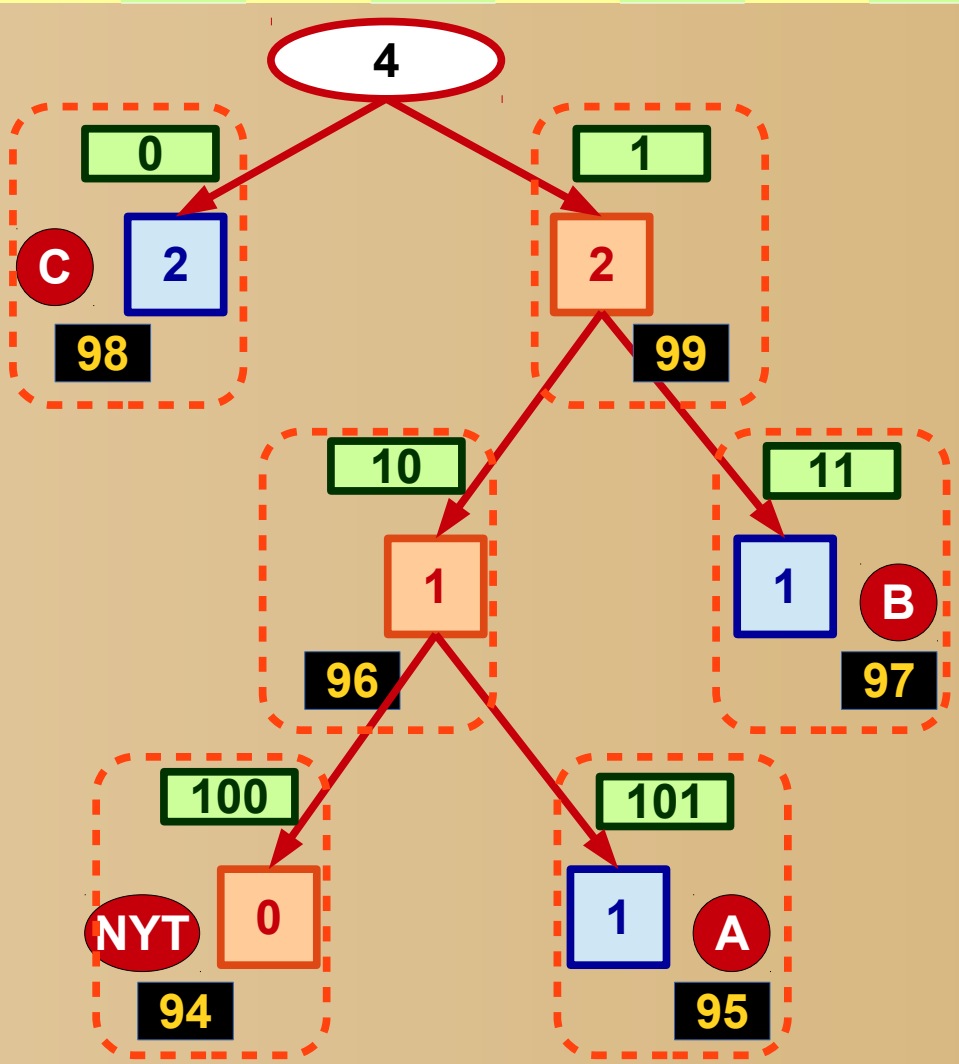
Prof. Khaled Mostafa
khaledms@fci-cu.edu.eg

Example: Adaptive Huffman Compression

جامعة القاهرة
كلية الحاسبات و المعلومات

Symbol	"A"	NYT	"B"	NYT	"C"	"C"									
Code	00	0	01	00	10	101									

Symbol	Short Code
A	00
B	01
C	10



ABCCCAAAA

Code

Update Tree

Swap: Done [95], [98]
Inc Counter: Node 98 {1+}
Goto Parent: Root,
Inc {3+}

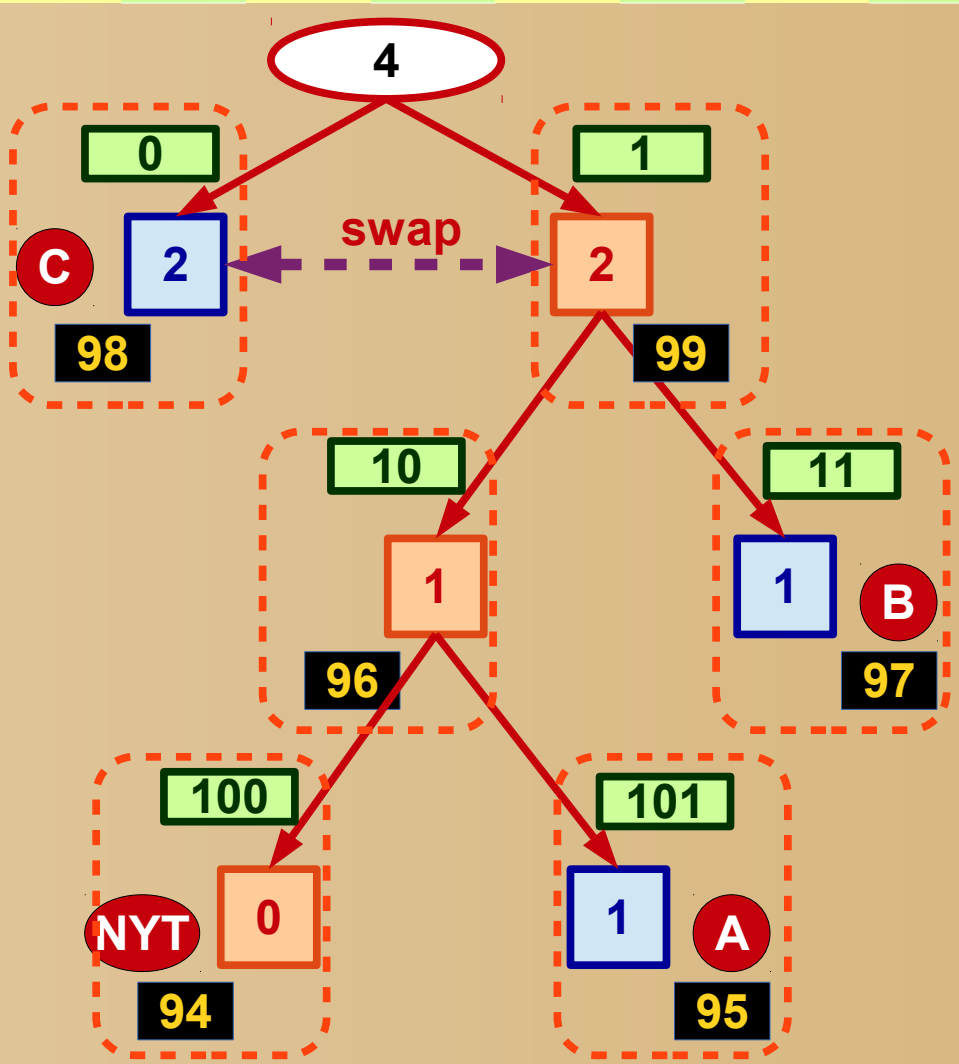
Prof. Khaled Mostafa
khaledms@fci-cu.edu.eg

Example: Adaptive Huffman Compression

جامعة القاهرة
كلية الحاسبات و المعلومات

Symbol	"A"	NYT	"B"	NYT	"C"	"C"	"C"								
Code	00	0	01	00	10	101	0								

Symbol	Short Code
A	00
B	01
C	10



ABCCCAAAA

Code

Symbol Occurred before:
Node: [98]
Code =[0]

Update Tree

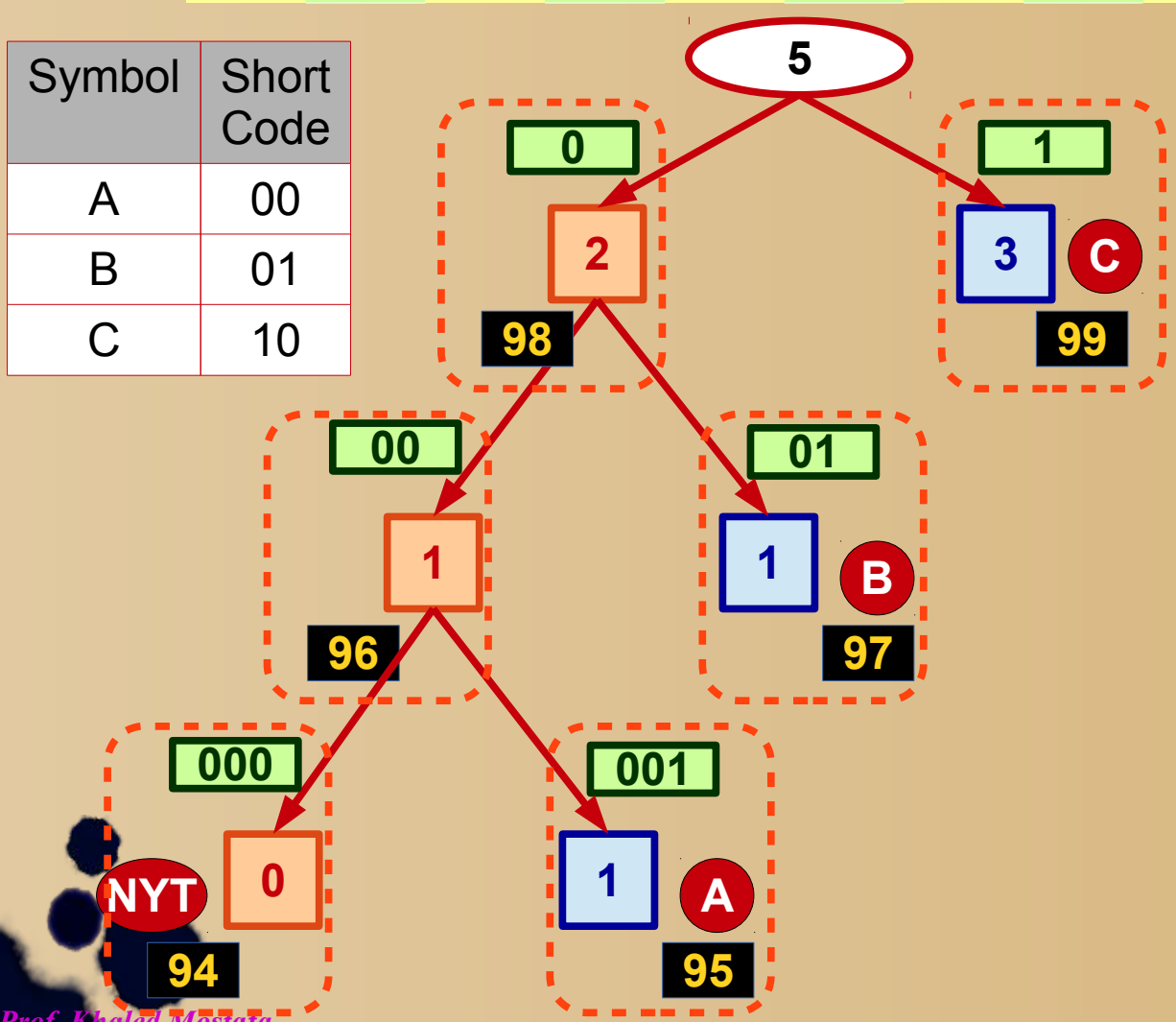
Go To Node: 98
Need Swap: YES [98] , [99]

Prof. Khaled Mostafa
khaledms@fci-cu.edu.eg

Example: Adaptive Huffman Compression

Symbol	"A"	NYT	"B"	NYT	"C"	"C"	"C"								
Code	00	0	01	00	10	101	0								

Symbol	Short Code
A	00
B	01
C	10



ABCCCAAAA

Code

Update Tree

Swap: Done [99, [98]

Inc Counter: Node 99 {2+}

Goto Parent: Root,

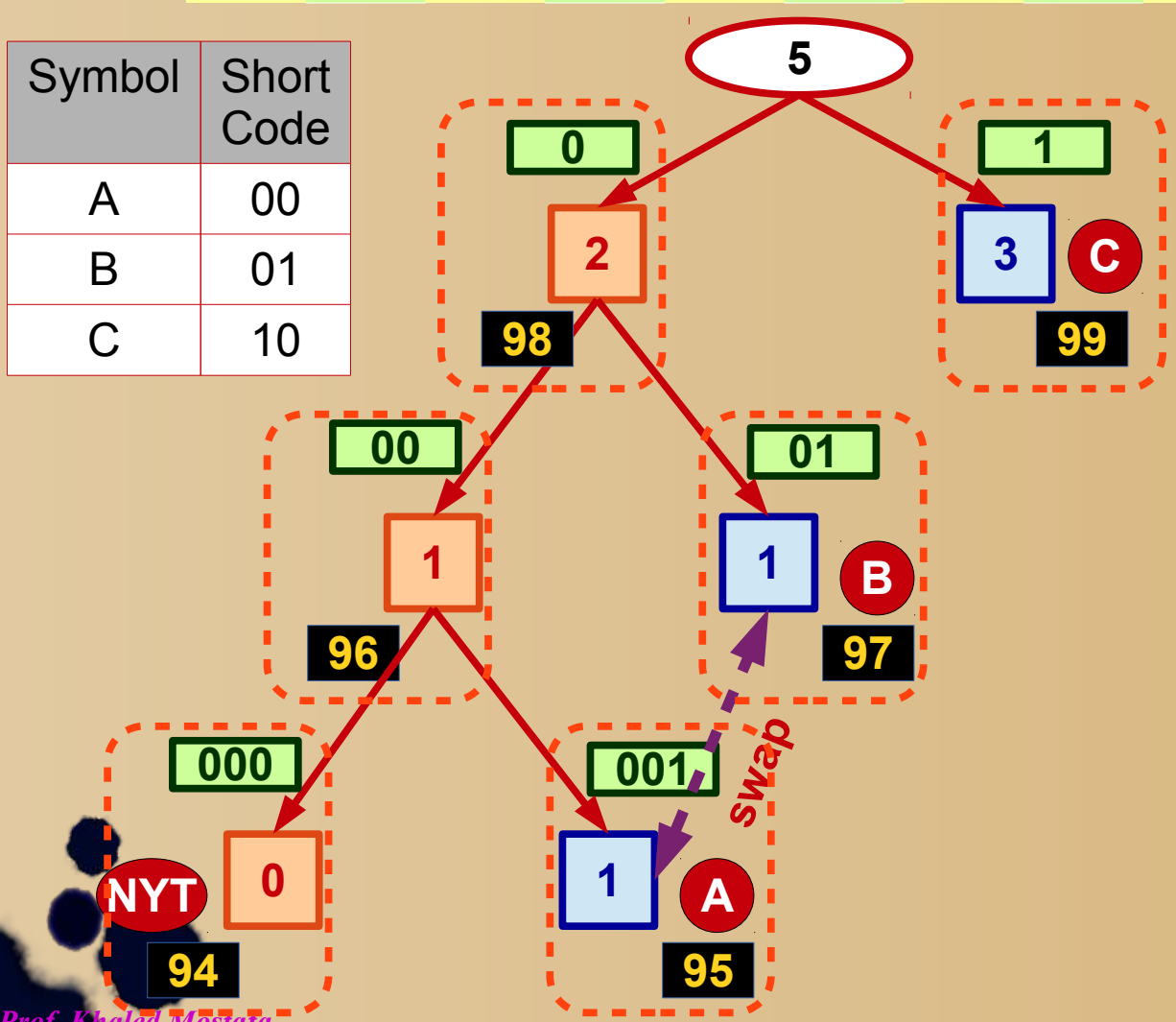
Inc {4+}

Prof. Khaled Mostafa
khaledms@fci-cu.edu.eg

Example: Adaptive Huffman Compression

Symbol	"A"	NYT	"B"	NYT	"C"	"C"	"C"	"A"							
Code	00	0	01	00	10	101	0	001							

Symbol	Short Code
A	00
B	01
C	10



ABCCCAAAA

Code

Symbol Occurred before:

Node: [95]

Code =[001]

Update Tree

Go To Node: 95

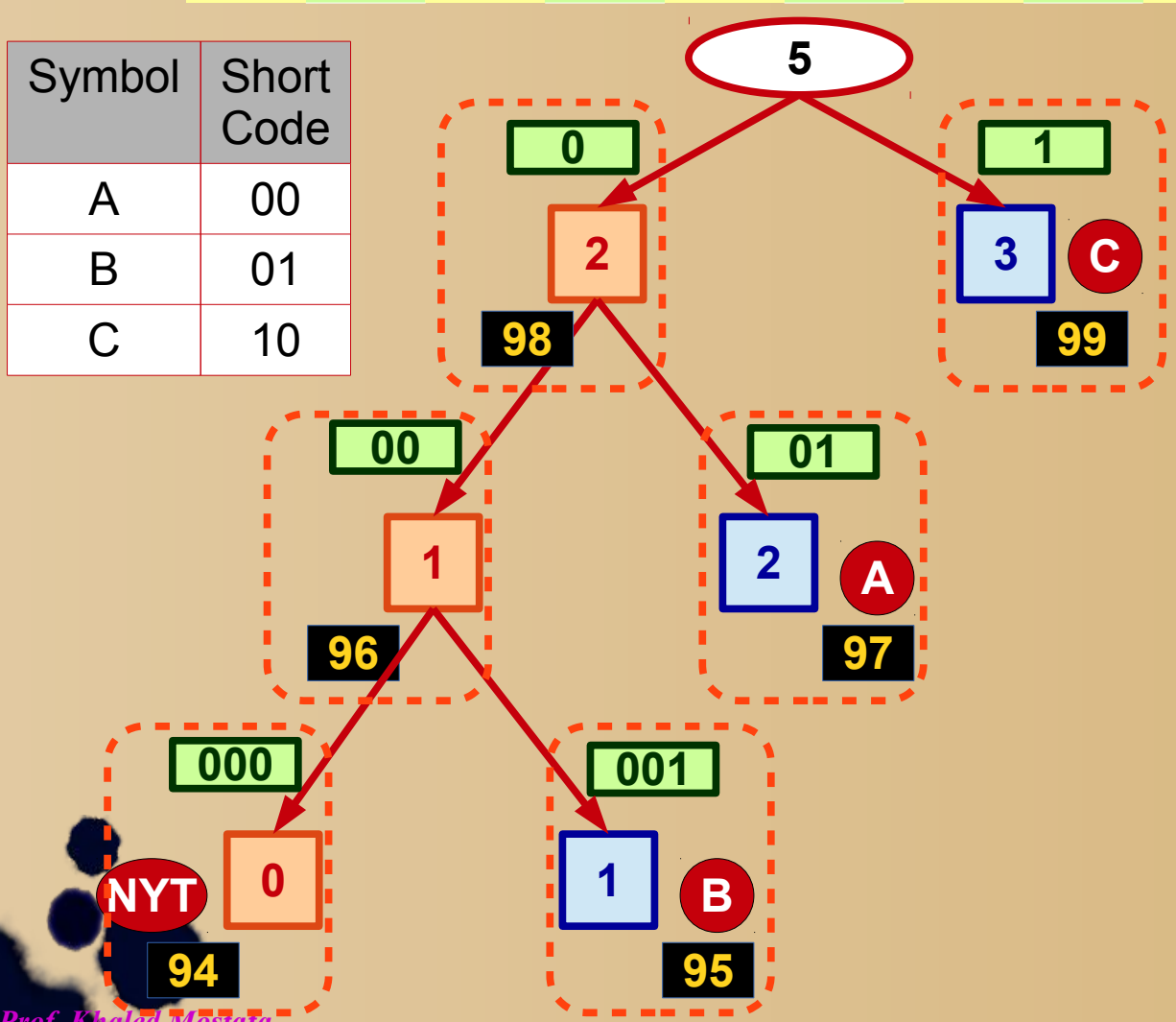
Need Swap: YES [95] , [97]

Prof. Khaled Mostafa
khaledms@fci-cu.edu.eg

Example: Adaptive Huffman Compression

Symbol	"A"	NYT	"B"	NYT	"C"	"C"	"C"	"A"
Code	00	0	01	00	10	101	0	001

Symbol	Short Code
A	00
B	01
C	10



ABCCCAAAA

Code

Update Tree

Swap: Done [95] , [97]

Inc Counter: Node 97 {1+}

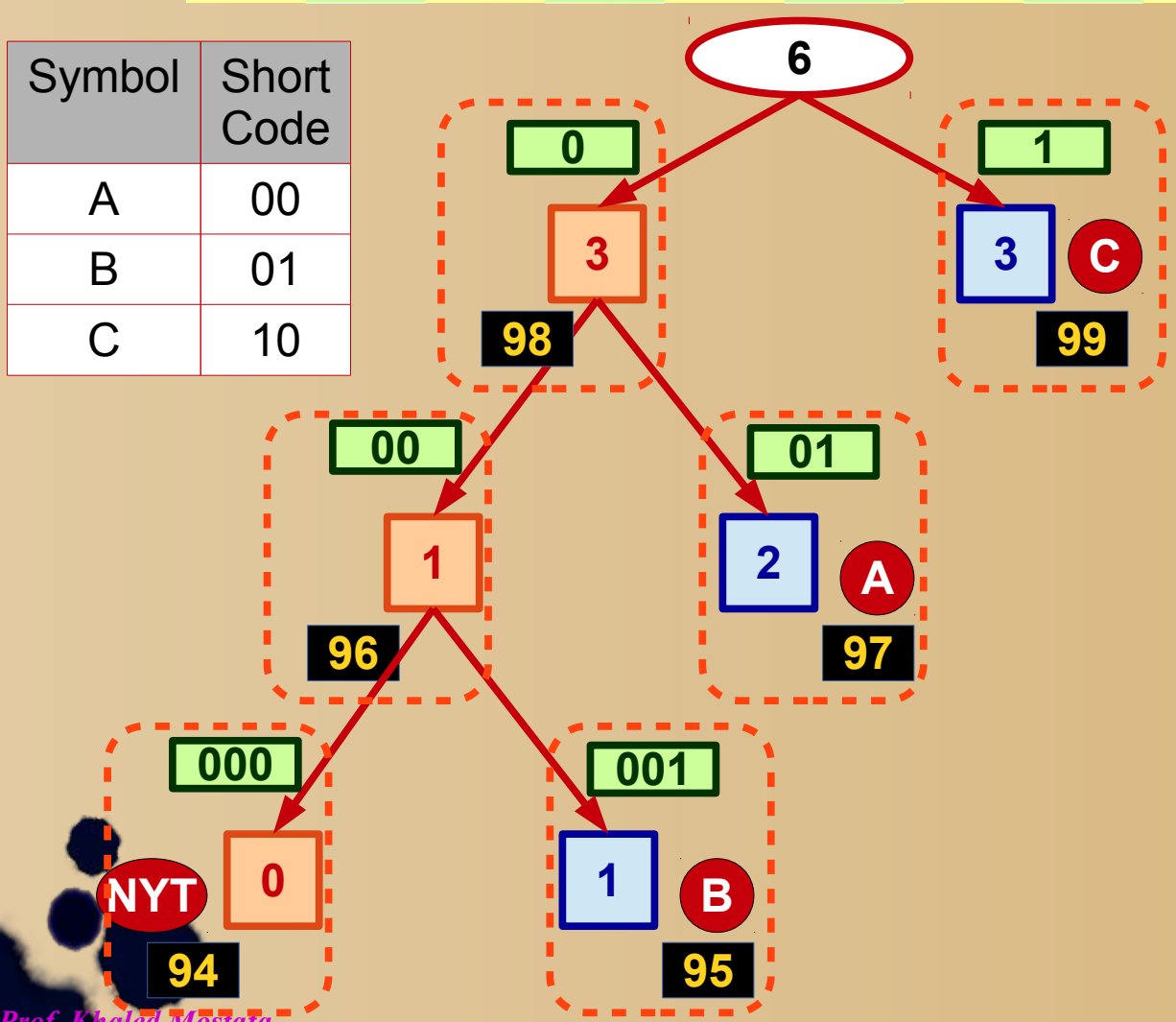
Goto Parent: Node [98]

Prof. Khaled Mostafa
khaledms@fci-cu.edu.eg

Example: Adaptive Huffman Compression

Symbol	"A"	NYT	"B"	NYT	"C"	"C"	"C"	"A"							
Code	00	0	01	00	10	101	0	001							

Symbol	Short Code
A	00
B	01
C	10



ABCCCAAAA

Code

Update Tree

Need Swap:No

Inc Counter: Node 98 {2+}

Goto Parent: Root,

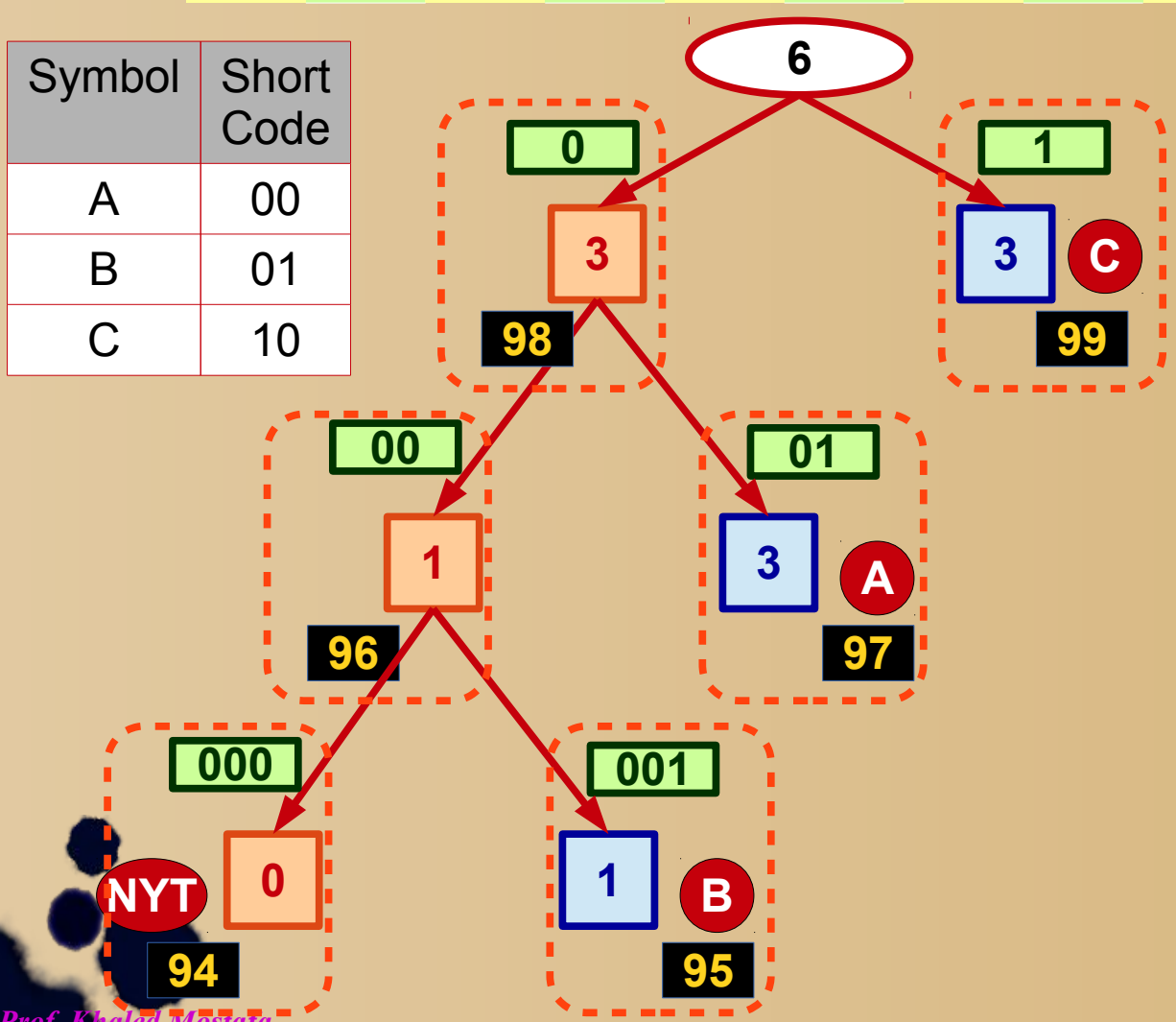
Inc {5+}

Prof. Khaled Mostafa
khaledms@fci-cu.edu.eg

Example: Adaptive Huffman Compression

Symbol	"A"	NYT	"B"	NYT	"C"	"C"	"C"	"A"	"A"
Code	00	0	01	00	10	101	0	001	01

Symbol	Short Code
A	00
B	01
C	10



ABCCCAAAA

Code

Symbol Occurred before:

Node: [97]

Code =[01]

Update Tree

Need Swap: No

Inc Counter: Node 97 {2+}

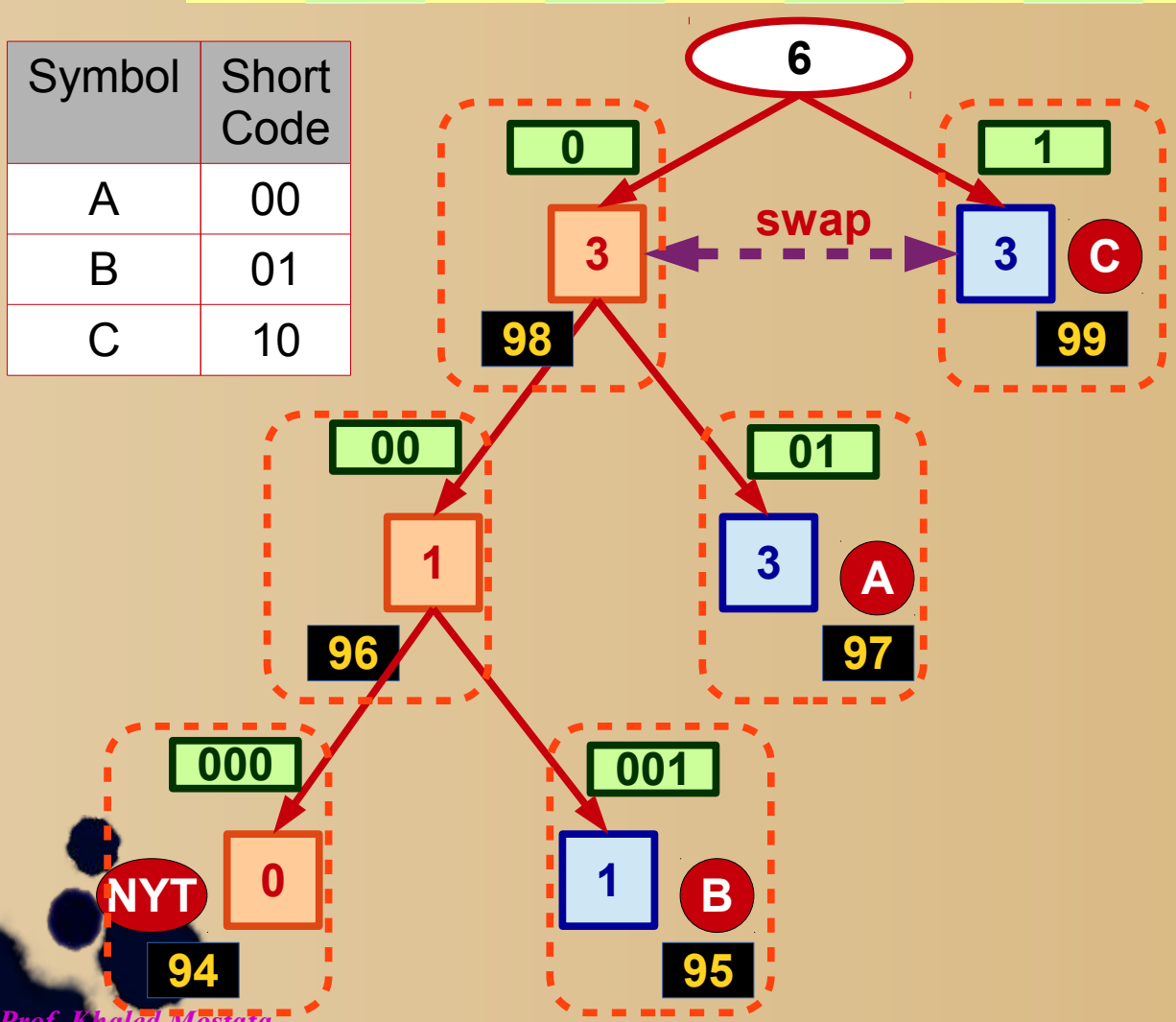
Goto Parent: Node [98]

Prof. Khaled Mostafa
khaledms@fci-cu.edu.eg

Example: Adaptive Huffman Compression

Symbol	"A"	NYT	"B"	NYT	"C"	"C"	"C"	"A"	"A"
Code	00	0	01	00	10	101	0	001	01

Symbol	Short Code
A	00
B	01
C	10



ABCCCAAAA

Code

Update Tree

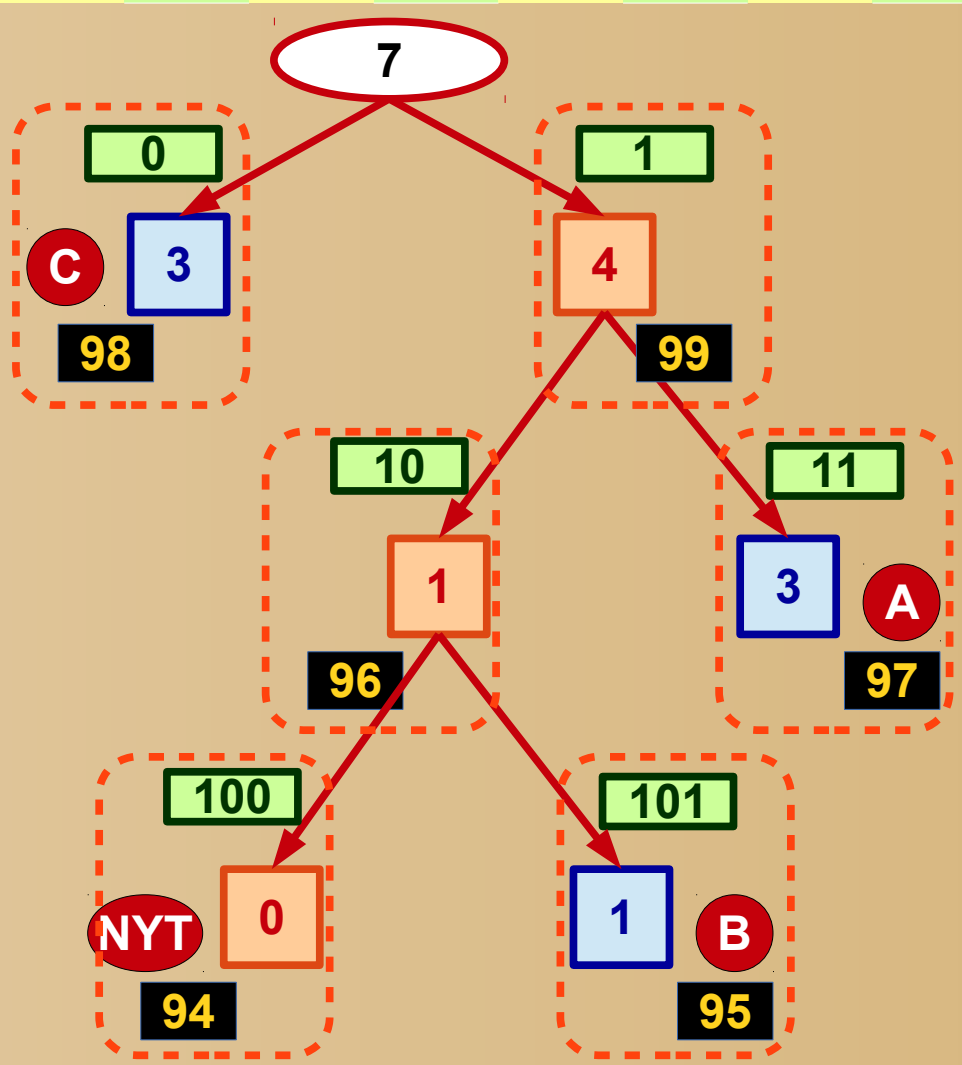
Need Swap: Yes [98] , [99]

Prof. Khaled Mostafa
khaledms@fci-cu.edu.eg

Example: Adaptive Huffman Compression

Symbol	"A"	NYT	"B"	NYT	"C"	"C"	"C"	"A"	"A"
Code	00	0	01	00	10	101	0	001	01

Symbol	Short Code
A	00
B	01
C	10



ABCCCAAAA

Code

Update Tree

Swap: Done [98] , [99]

Inc Counter: Node 99 {3+}

Goto Parent: Root,

Inc {6+}

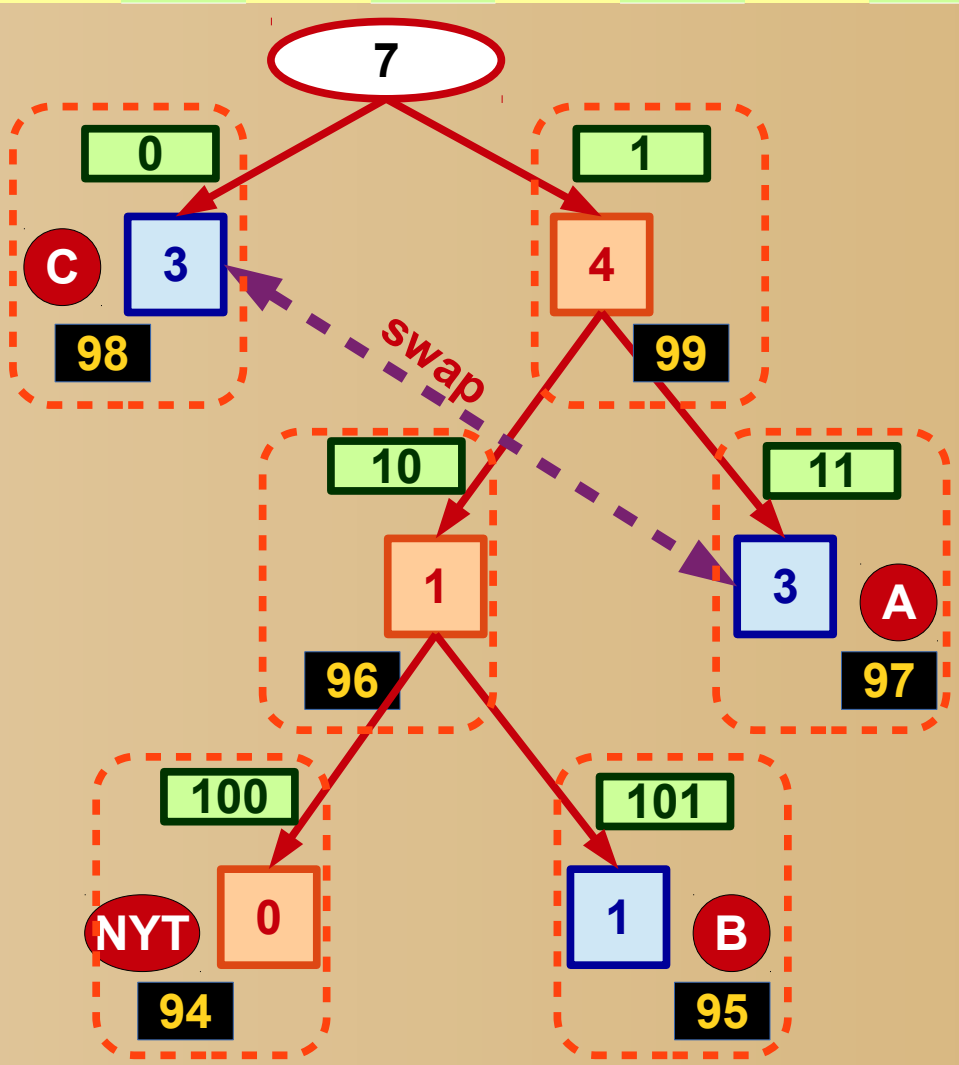
Prof. Khaled Mostafa
khaledms@fci-cu.edu.eg

Example: Adaptive Huffman Compression

جامعة القاهرة
كلية الحاسبات
و المعلومات

Symbol	"A"	NYT	"B"	NYT	"C"	"C"	"C"	"A"	"A"	"A"
Code	00	0	01	00	10	101	0	001	01	11

Symbol	Short Code
A	00
B	01
C	10



ABCCCAAAA

Code

Symbol Occurred before:
Node: [97]
Code =[11]

Update Tree

Go To Node: 97
Need Swap: YES [97] , [98]

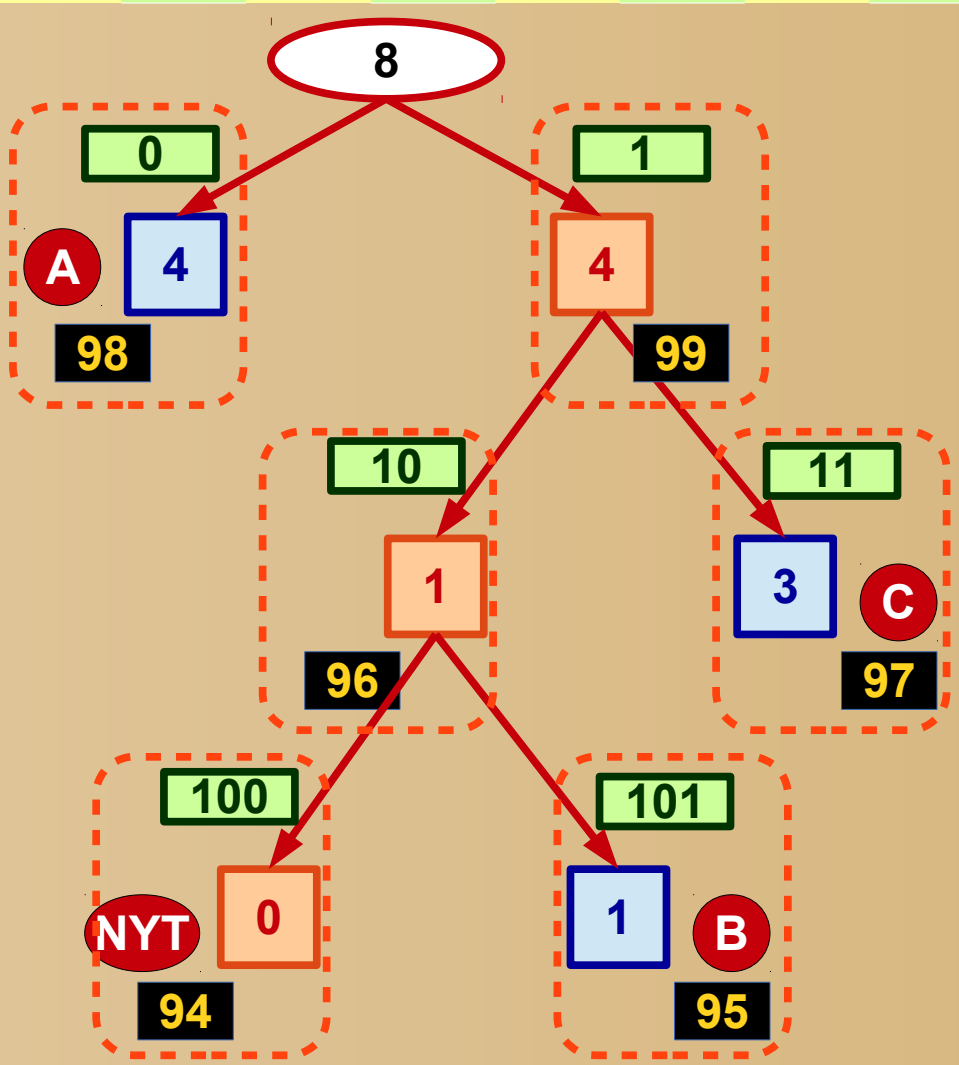
Prof. Khaled Mostafa
khaledms@fci-cu.edu.eg

Example: Adaptive Huffman Compression

جامعة القاهرة
كلية الحاسبات و المعلومات

Symbol	"A"	NYT	"B"	NYT	"C"	"C"	"C"	"A"	"A"	"A"
Code	00	0	01	00	10	101	0	001	01	11

Symbol	Short Code
A	00
B	01
C	10



ABCCCAAAA

Code

Update Tree

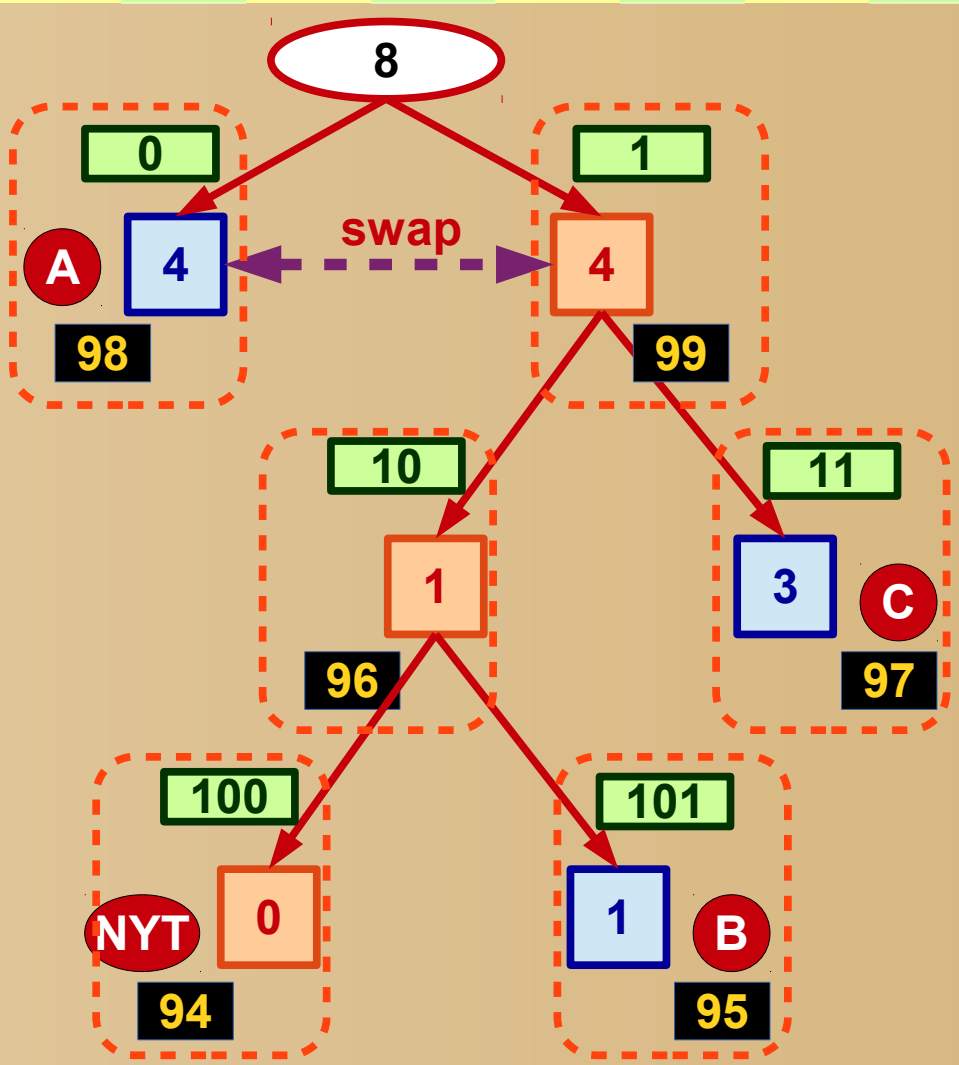
Swap: Done [98] , [97]
Inc Counter: Node 98 {3+}
Goto Parent: Root,
Inc {7+}

Prof. Khaled Mostafa
khaledms@fci-cu.edu.eg

Example: Adaptive Huffman Compression

Symbol	"A"	NYT	"B"	NYT	"C"	"C"	"C"	"A"	"A"	"A"	"A"				
Code	00	0	01	00	10	101	0	001	01	11	0				

Symbol	Short Code
A	00
B	01
C	10



ABCCCAAAA

Code

Symbol Occurred before:

Node: [98]

Code =[0]

Update Tree

Go To Node: 98

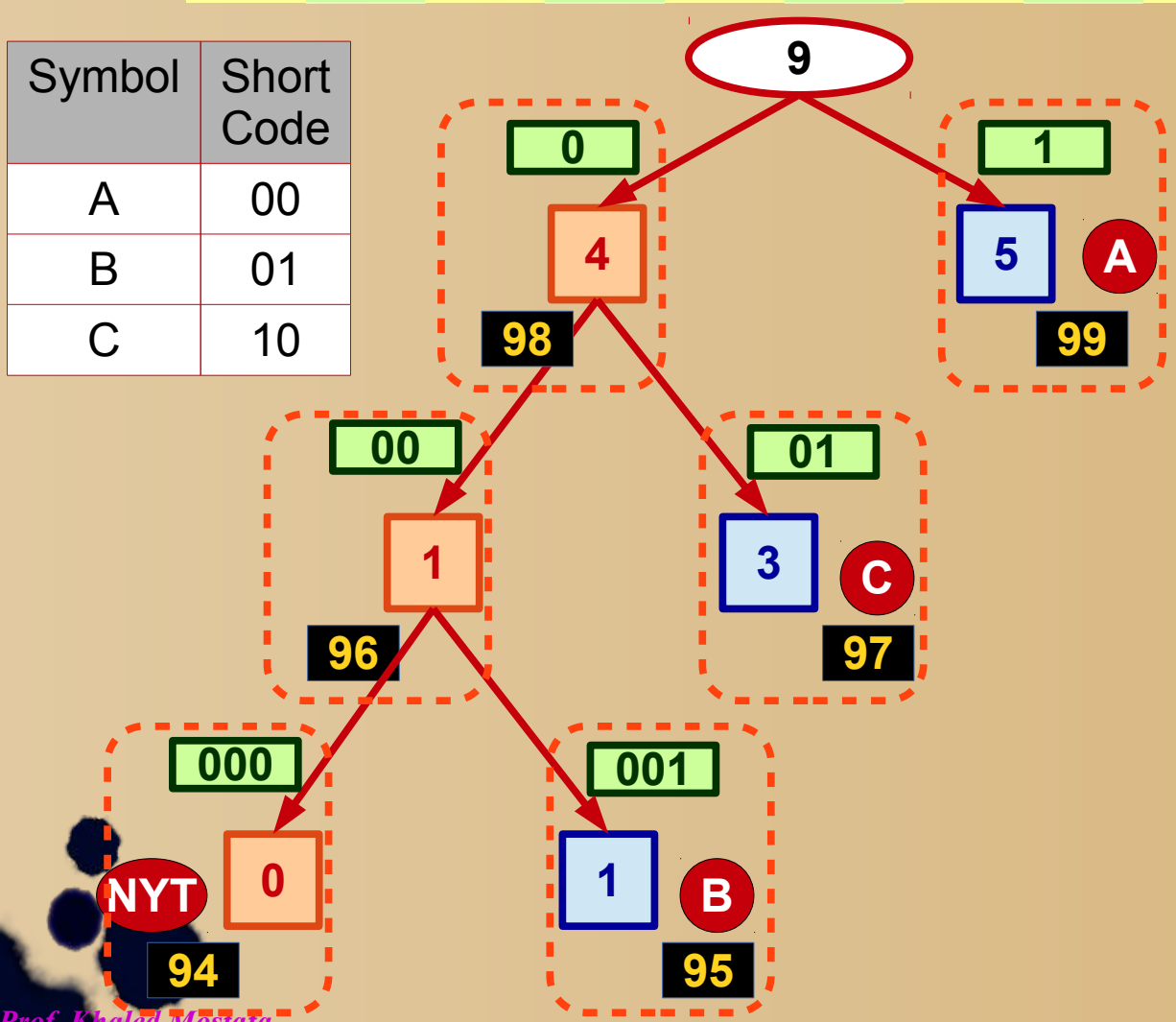
Need Swap: YES [98] , [99]

Prof. Khaled Mostafa
khaledms@fci-cu.edu.eg

Example: Adaptive Huffman Compression

Symbol	"A"	NYT	"B"	NYT	"C"	"C"	"C"	"A"	"A"	"A"	"A"				
Code	00	0	01	00	10	101	0	001	01	11	0				

Symbol	Short Code
A	00
B	01
C	10



ABCCCAAAA

Code

Update Tree

Swap: Done [98] , [99]

Inc Counter: Node 99 {5+}

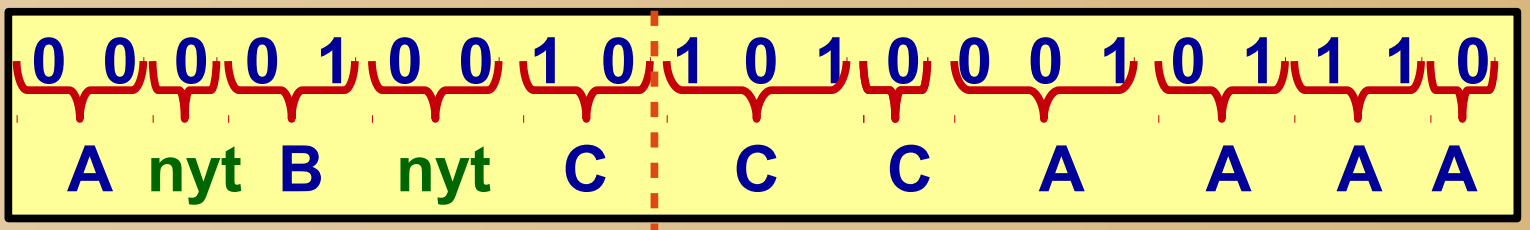
Goto Parent: Root,

Inc {8+}

Prof. Khaled Mostafa
khaledms@fci-cu.edu.eg

Example: Adaptive Huffman Compression

Compressed Code



Symbol	NTY Code	Short Code	Huffman Code	Number of Bits
A	----	00	----	2
B	0	01	----	3
C	00	10	----	4
C			101	3
C			0	1
A			001	3
A			01	2
A			11	2
A			0	1

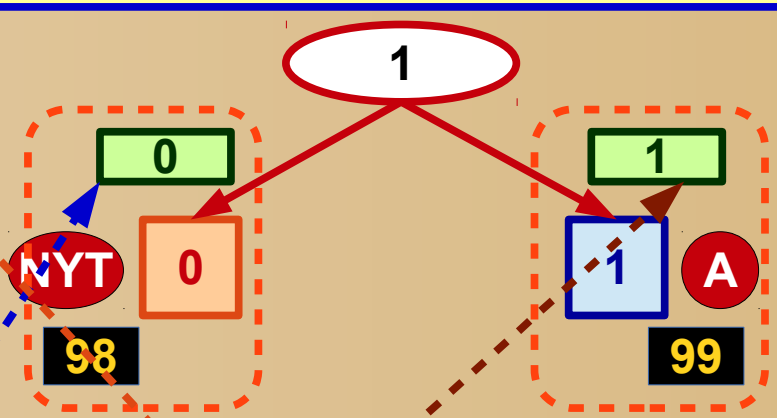
Prof. Khaled Mostafa
khaledms@fci-cu.edu.eg

Example: Adaptive Huffman De-Compression

0 0 0 0 1 0 0 1 0 1 0 1 0 0 0 1 0 1 1 1 0

A

Symbol	Short Code
A	00
B	01
C	10



A

Parse Code

Short Code of First Symbol

Code = [00]

Symbol: "A"

How to Parse Code

Parse Code till get either

"NTY" Code or

"Node Symbol" Code

If it is "NYT" then next bits are Short Codes of Symbol

Update Tree

Go To Node: Root

Split: Yes

Inc Counter: Node 99,Root

Goto Parent: Root

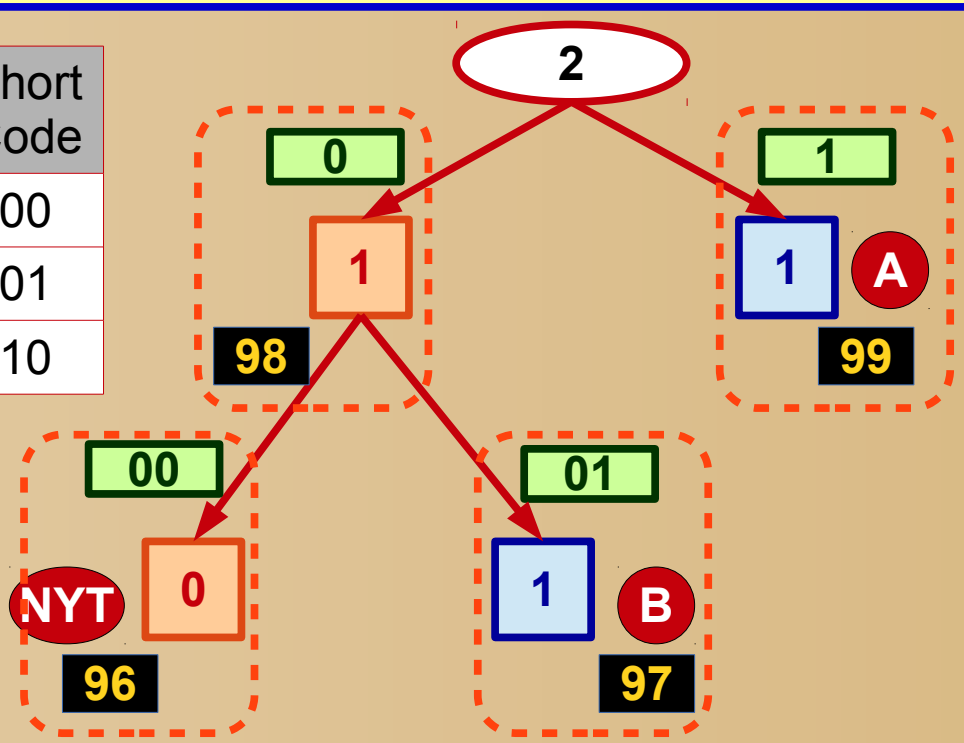
Prof. Khaled Mostafa
khaledms@fci-cu.edu.eg

Example: Adaptive Huffman De-Compression

0 0 0 0 1 0 0 1 0 1 0 1 0 0 0 1 0 1 1 1 0

A nyt B

Symbol	Short Code
A	00
B	01
C	10



AB

Parse Code

Code = [0] → NYT,
[01] "B"Short Code

Update Tree

Go To Node: NYT [98]
Split: Yes
Inc Counter: Node 97, 98
Goto Parent: Root,
Inc {1+}

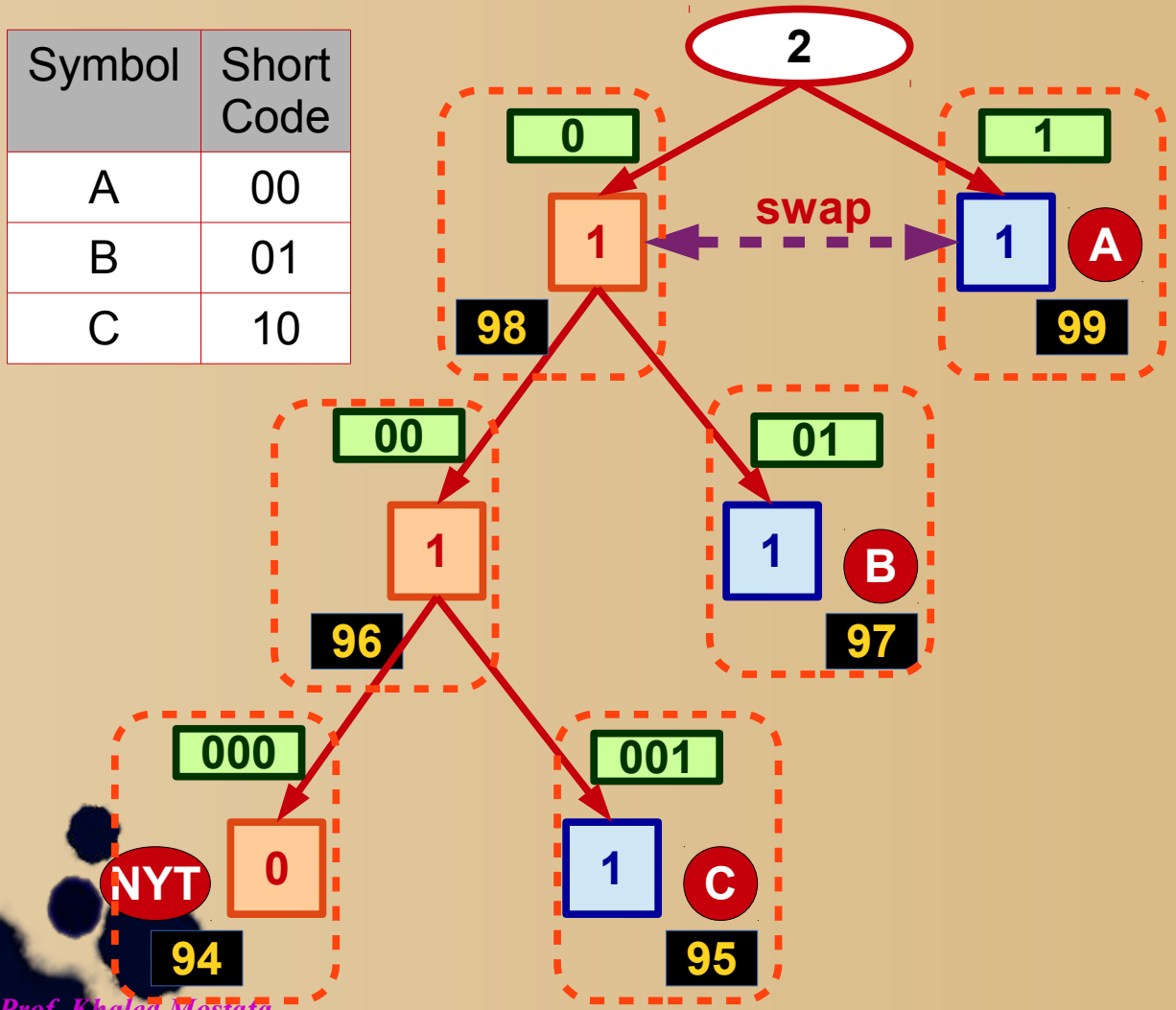
Prof. Khaled Mostafa
khaledms@fci-cu.edu.eg

Example: Adaptive Huffman De-Compression

0 0 0 0 1 0 0 1 0 1 0 1 0 0 0 1 0 1 1 1 0

A nyt B nyt C

Symbol	Short Code
A	00
B	01
C	10



ABC

Parse Code

Code = [00] → NYT,
[1] "C"Short Code

Update Tree

Go To Node: NYT [96]
Split: Yes
Inc Counter: Node 95, 96
Goto Parent: 98
Need Swap: Yes [98], [99]

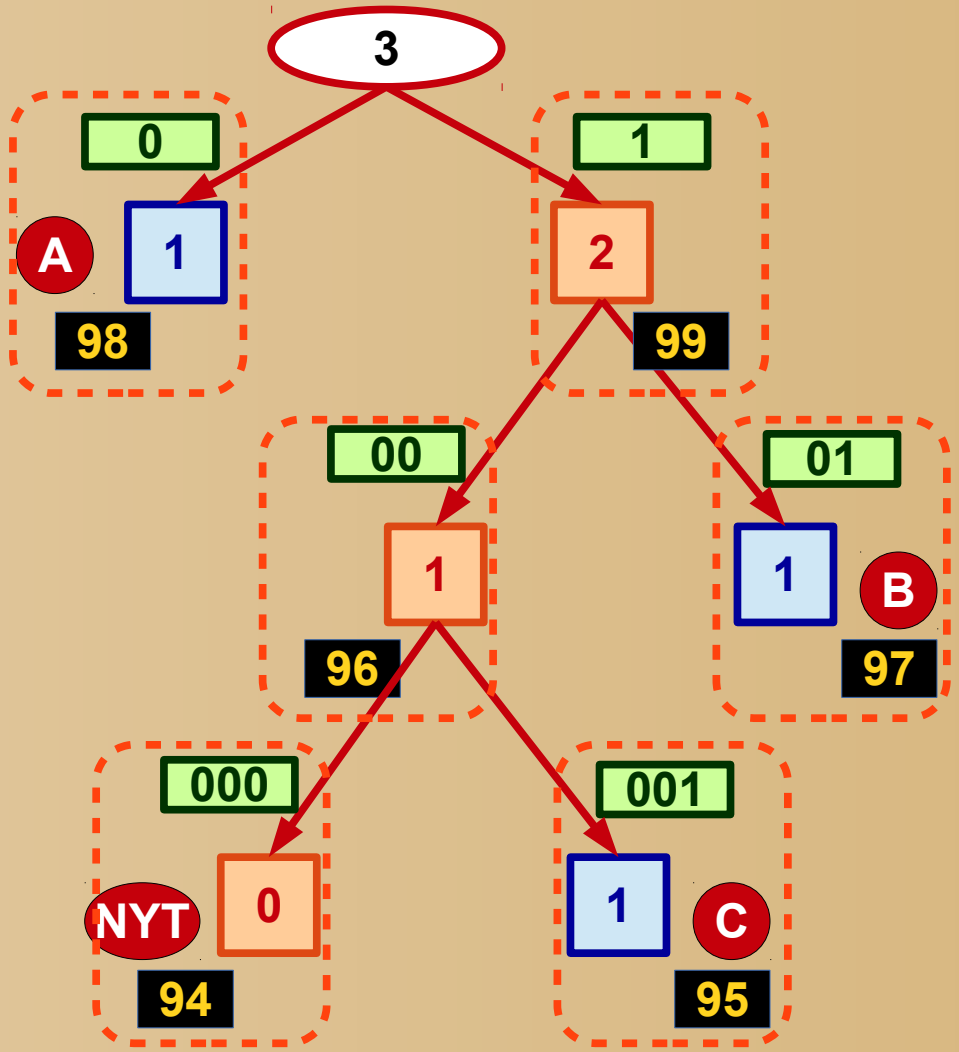
Prof. Khaled Mostafa
khaledms@fci-cu.edu.eg

Example: Adaptive Huffman De-Compression

0 0 0 0 1 0 0 1 0 1 0 1 0 0 0 1 0 1 1 1 0

A nyt B nyt C

Symbol	Short Code
A	00
B	01
C	10



ABC

Parse Code

Update Tree

Swap: Done [98], [99]
Inc Counter: Node 99 [1+]
Goto Parent: Root,
Inc {2+}

Prof. Khaled Mostafa
khaledms@fci-cu.edu.eg

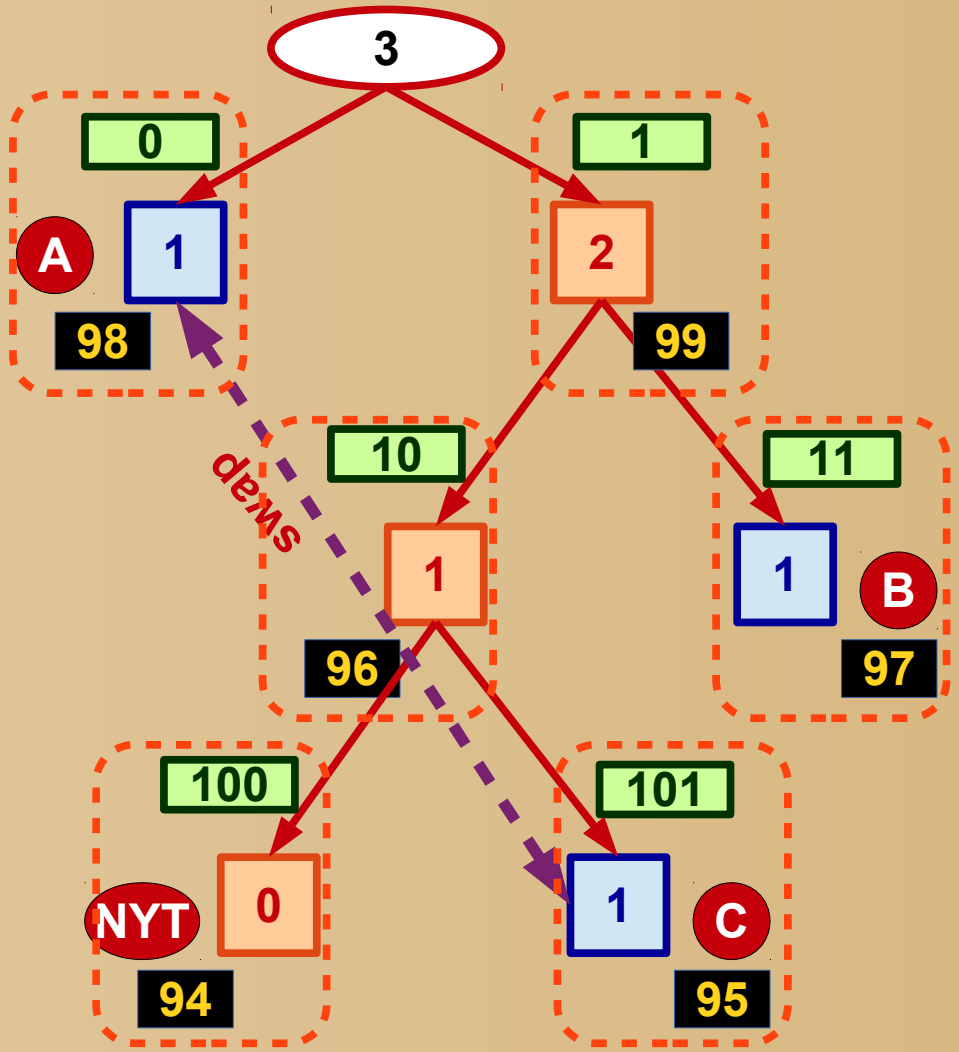
Example: Adaptive Huffman De-Compression

جامعة القاهرة
كلية الحاسبات و المعلومات

0 0 0 0 1 0 0 1 0 1 0 1 0 0 0 1 0 1 1 1 0

A nyt B nyt C C

Symbol	Short Code
A	00
B	01
C	10



ABCC

Parse Code

Code = [101] → "C"

Update Tree

Go To Node: 95

Need Swap: YES [95], [98]

Prof. Khaled Mostafa
khaledms@fci-cu.edu.eg

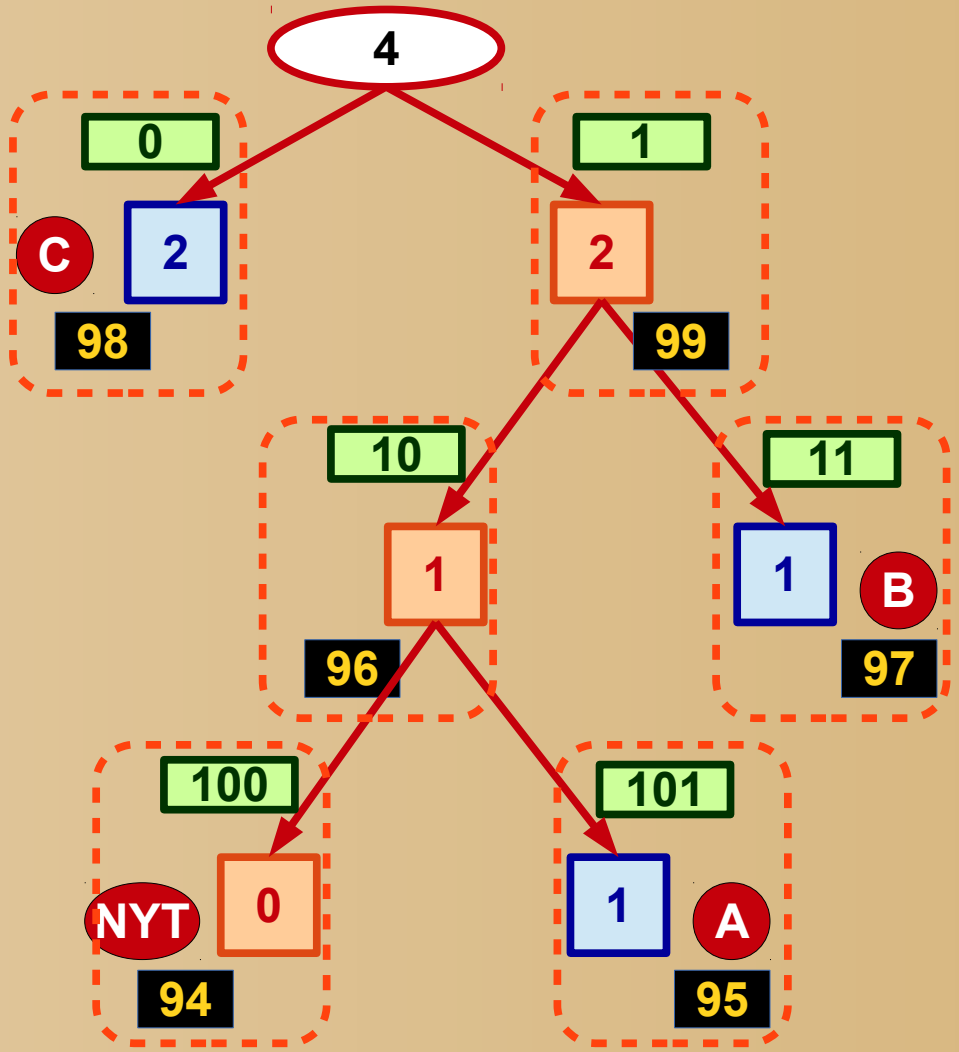
Example: Adaptive Huffman De-Compression

جامعة القاهرة
كلية الحاسبات و المعلومات

0 0 0 0 1 0 0 1 0 1 0 1 0 0 0 1 0 1 1 1 0

A nyt B nyt C C

Symbol	Short Code
A	00
B	01
C	10



ABCC

Parse Code

Update Tree

Swap: Done [95], [98]

Inc Counter: Node 98 {1+}

Goto Parent: Root,
Inc {3+}

Prof. Khaled Mostafa
khaledms@fci-cu.edu.eg

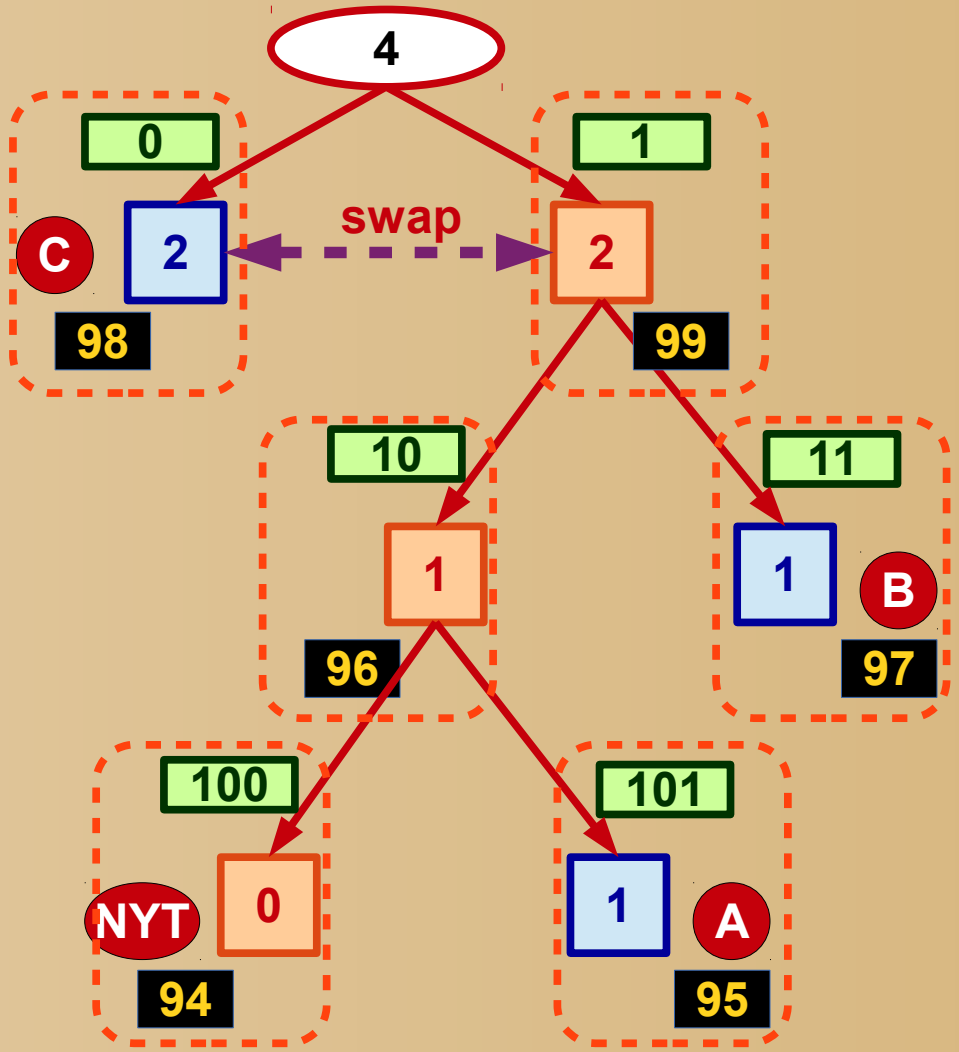
Example: Adaptive Huffman De-Compression

جامعة القاهرة
كلية الحاسبات و المعلومات

0 0 0 0 1 0 0 1 0 1 0 1 0 0 0 1 0 1 1 1 0

A nyt B nyt C C C

Symbol	Short Code
A	00
B	01
C	10



ABCCC

Parse Code

Code = [0] → "C"

Update Tree

Go To Node: 98

Need Swap: YES [98] , [99]

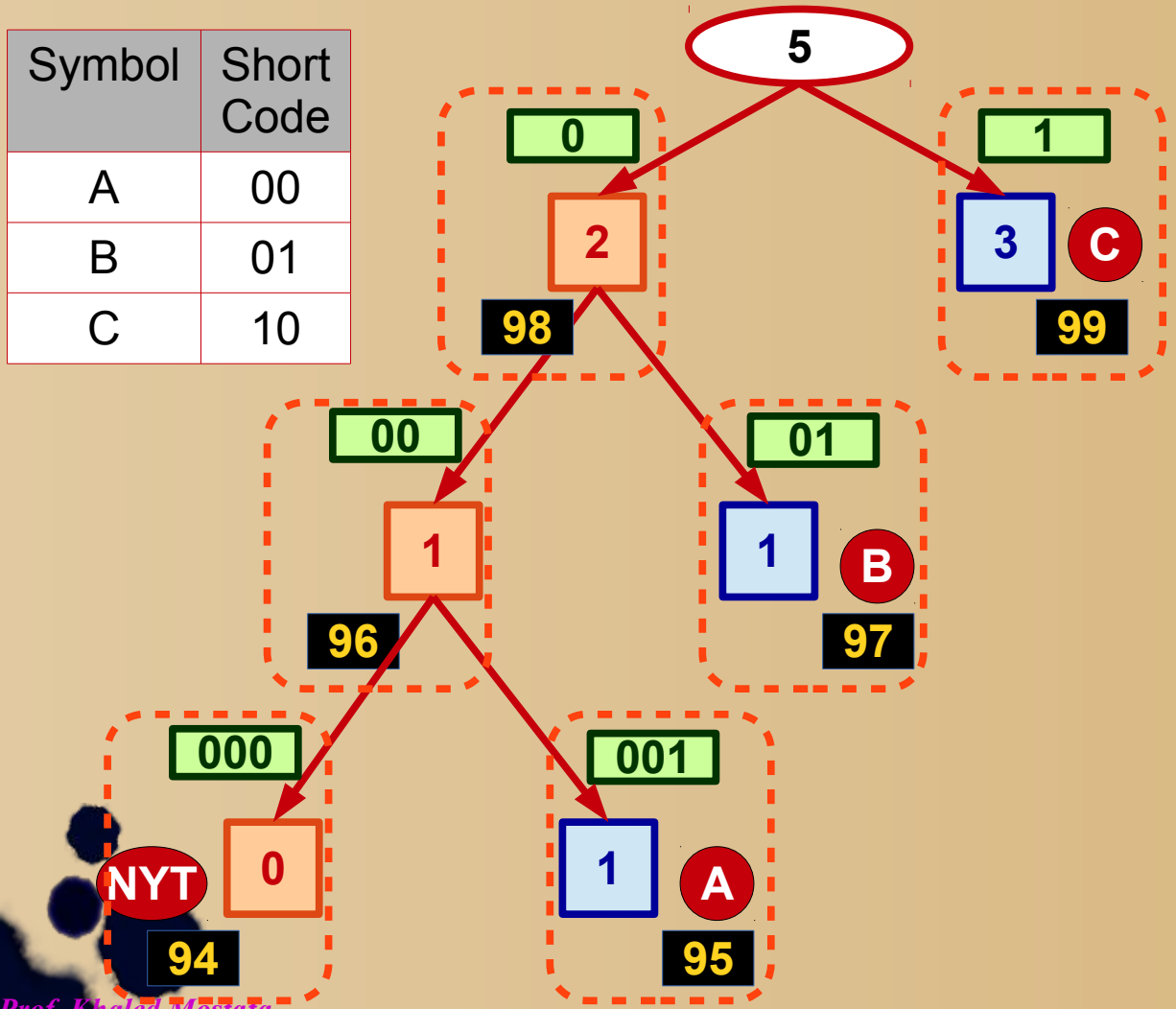
Prof. Khaled Mostafa
khaledms@fci-cu.edu.eg

Example: Adaptive Huffman De-Compression

0 0 0 0 1 0 0 1 0 1 0 1 0 0 0 1 0 1 1 1 0

A nyt B nyt C C C

Symbol	Short Code
A	00
B	01
C	10



ABCCC

Parse Code

Update Tree

Swap: Done [99, [98]

Inc Counter: Node 99 {2+}

Goto Parent: Root,

Inc {4+}

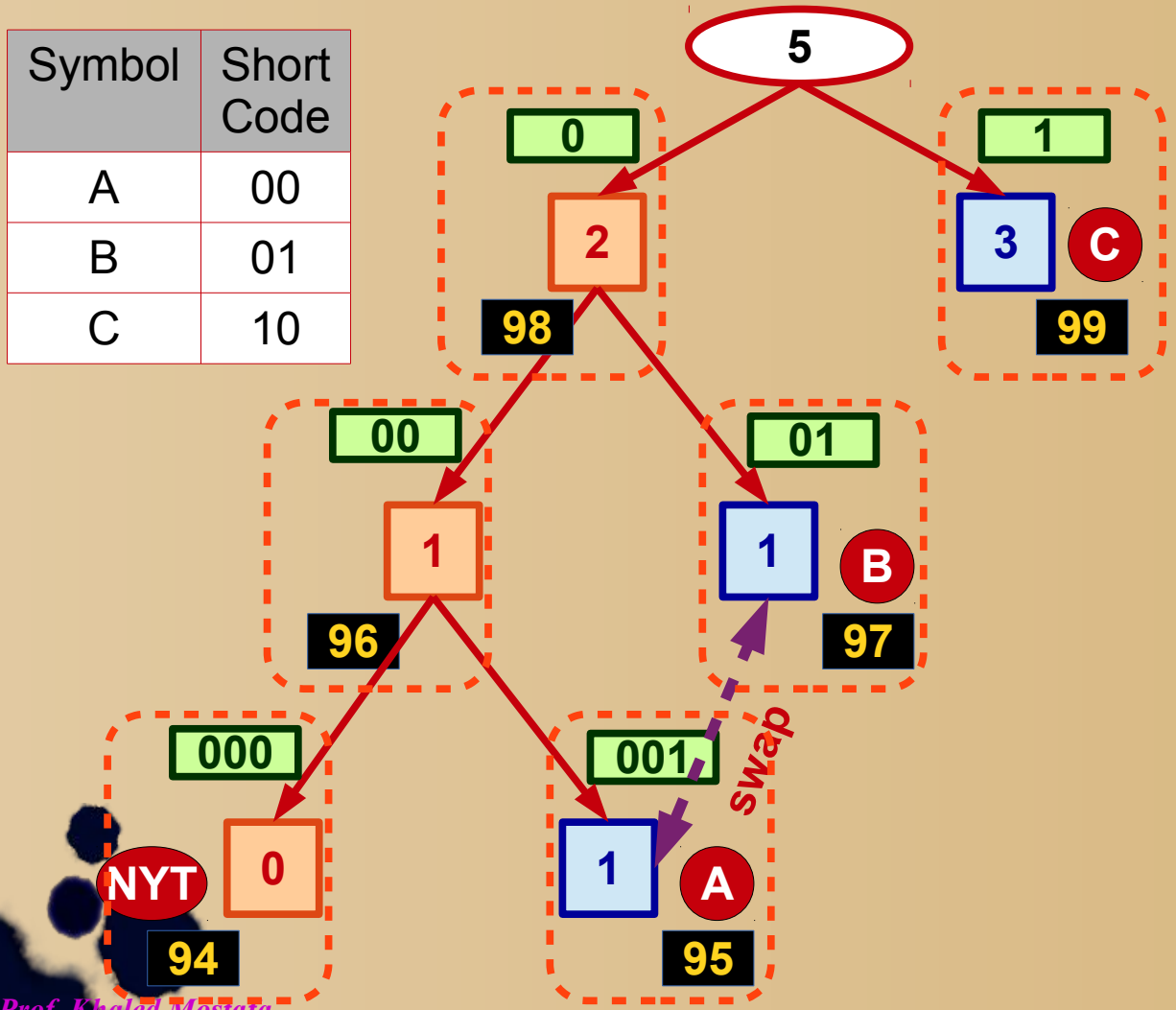
Prof. Khaled Mostafa
khaledms@fci-cu.edu.eg

Example: Adaptive Huffman De-Compression

0 0 0 0 1 0 0 1 0 1 0 1 0 0 0 1 0 1 1 1 0

A nyt B nyt C C C A

Symbol	Short Code
A	00
B	01
C	10



ABCCCA

Parse Code

Code = [101] → "A"

Update Tree

Go To Node: 95

Need Swap: YES [95] , [97]

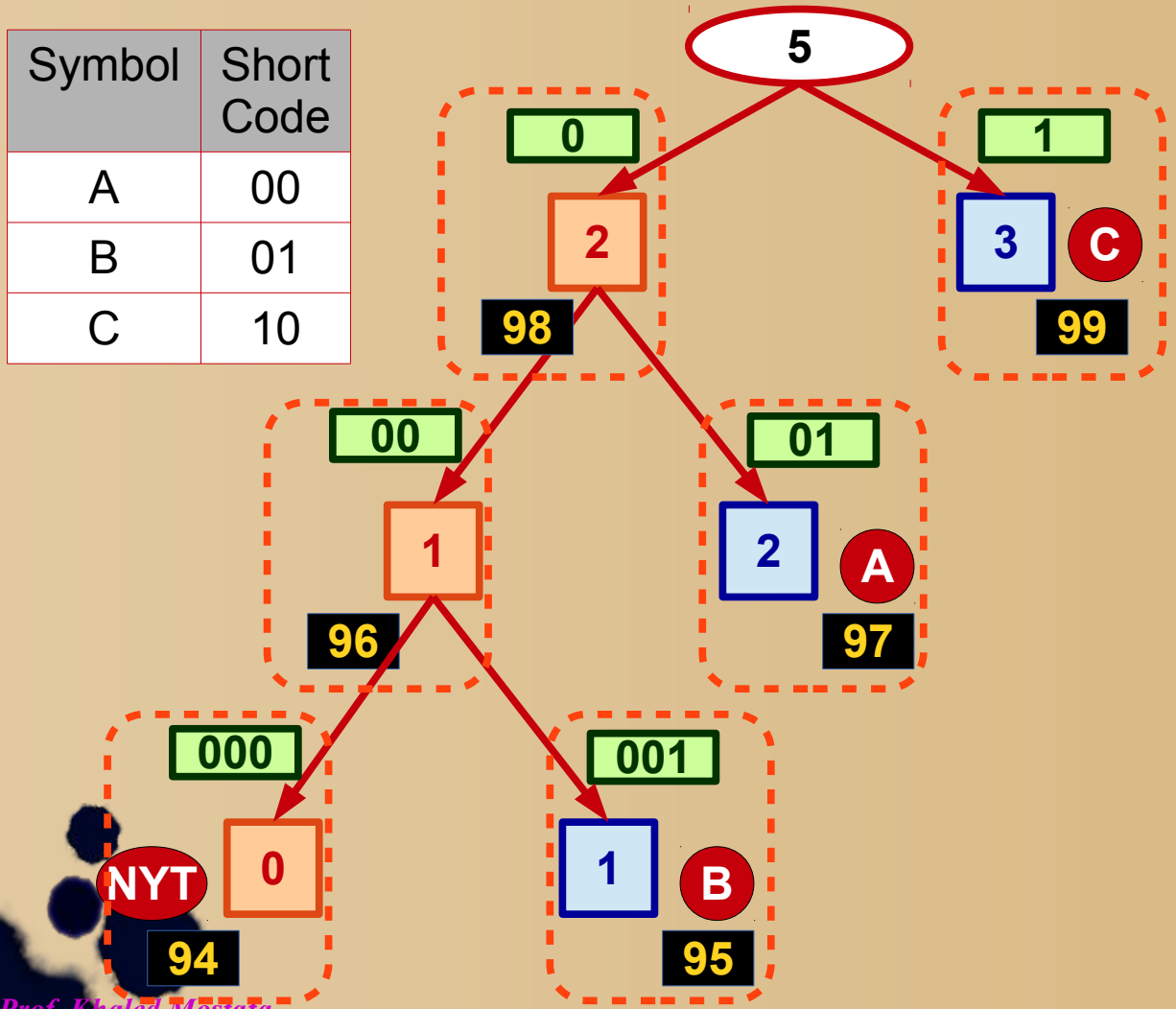
Prof. Khaled Mostafa
khaledms@fci-cu.edu.eg

Example: Adaptive Huffman De-Compression

0 0 0 0 1 0 0 1 0 1 0 1 0 0 0 1 0 1 1 1 0

A nyt B nyt C C C A

Symbol	Short Code
A	00
B	01
C	10



ABCCCA

Parse Code

Update Tree

Swap: Done [95] , [97]

Inc Counter: Node 97 {1+}

Goto Parent: Node [98]

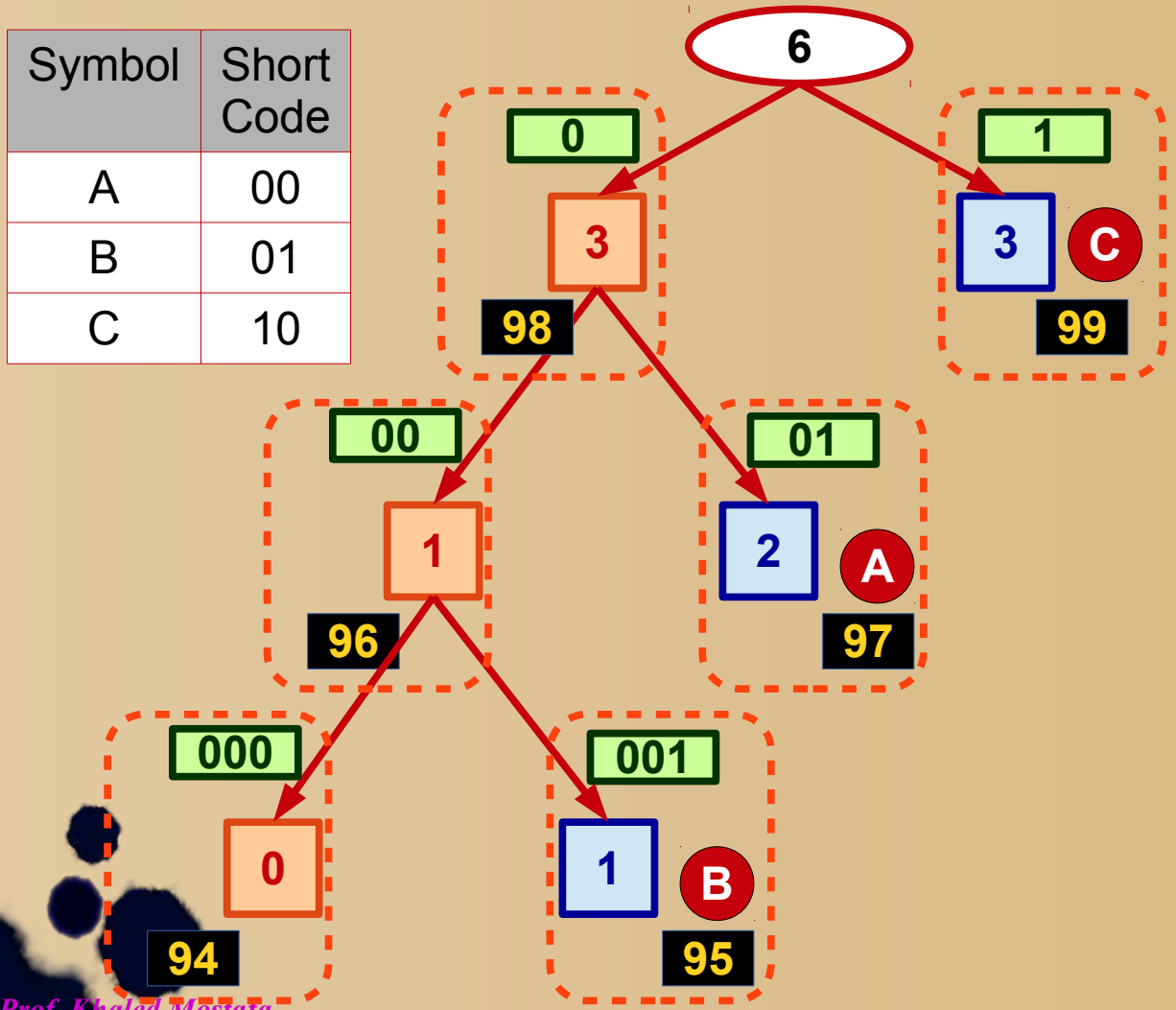
Prof. Khaled Mostafa
khaledms@fci-cu.edu.eg

Example: Adaptive Huffman De-Compression

0 0 0 0 1 0 0 1 0 1 0 1 0 0 0 1 0 1 1 1 0

A nyt B nyt C C C A

Symbol	Short Code
A	00
B	01
C	10



ABCCCA

Parse Code

Update Tree

Need Swap:No

Inc Counter: Node 98 {2+}

Goto Parent: Root,

Inc {5+}

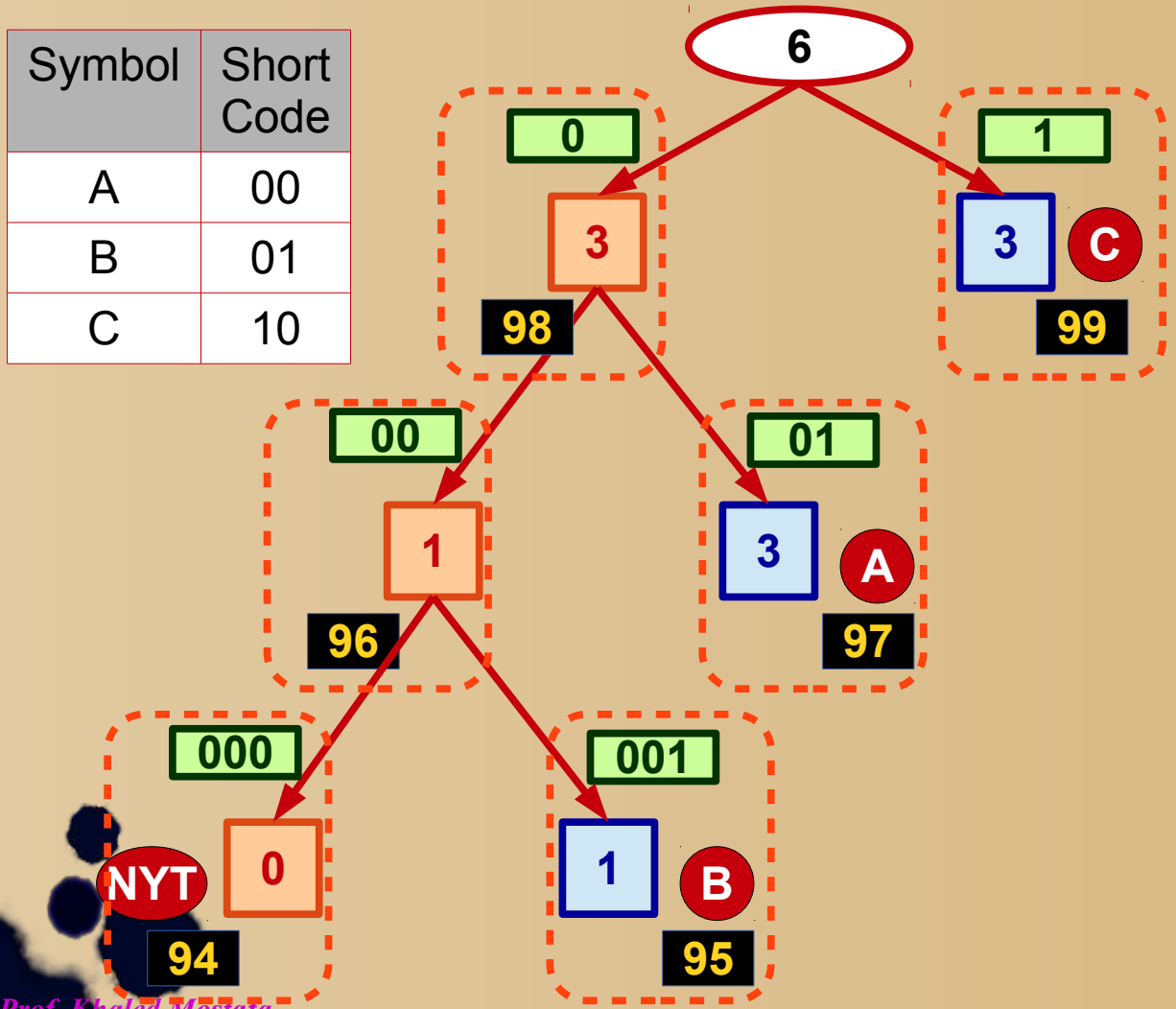
Prof. Khaled Mostafa
khaledms@fci-cu.edu.eg

Example: Adaptive Huffman De-Compression

0 0 0 0 1 0 0 1 0 1 0 1 0 0 0 1 0 1 1 1 0

A nyt B nyt C C C A A

Symbol	Short Code
A	00
B	01
C	10



ABCCCAA

Parse Code

Code = [01] → "A"

Update Tree

Need Swap:No

Inc Counter: Node 97 {2+}

Goto Parent: Node [98]

Prof. Khaled Mostafa
khaledms@fci-cu.edu.eg

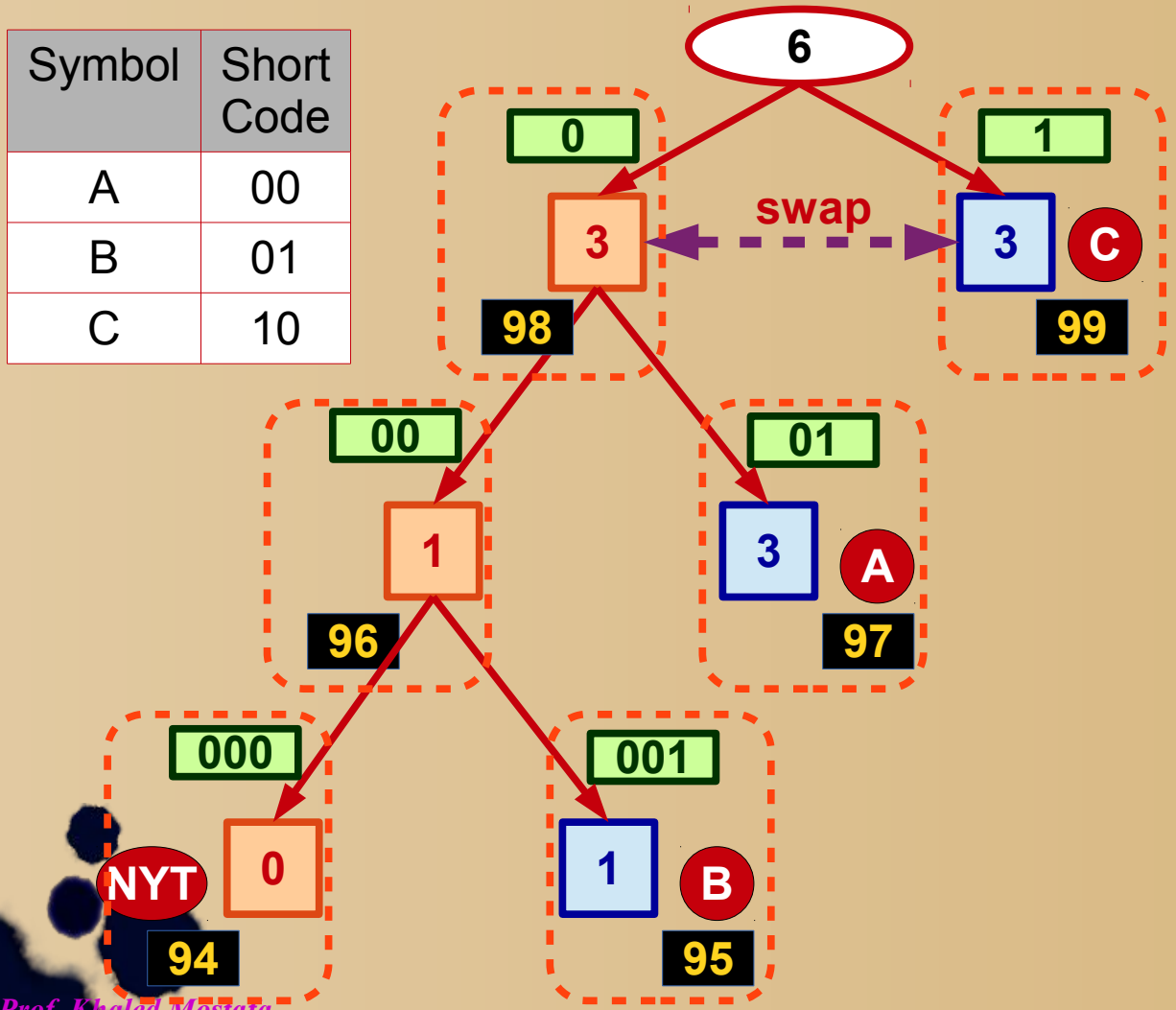
Example: Adaptive Huffman De-Compression

جامعة القاهرة
كلية الحاسبات و المعلومات

0 0 0 0 1 0 0 1 0 1 0 1 0 0 0 1 0 1 1 1 0

A nyt B nyt C C C A A

Symbol	Short Code
A	00
B	01
C	10



ABCCCAA

Parse Code

Update Tree

Need Swap: Yes [98] , [99]

Prof. Khaled Mostafa
khaledms@fci-cu.edu.eg

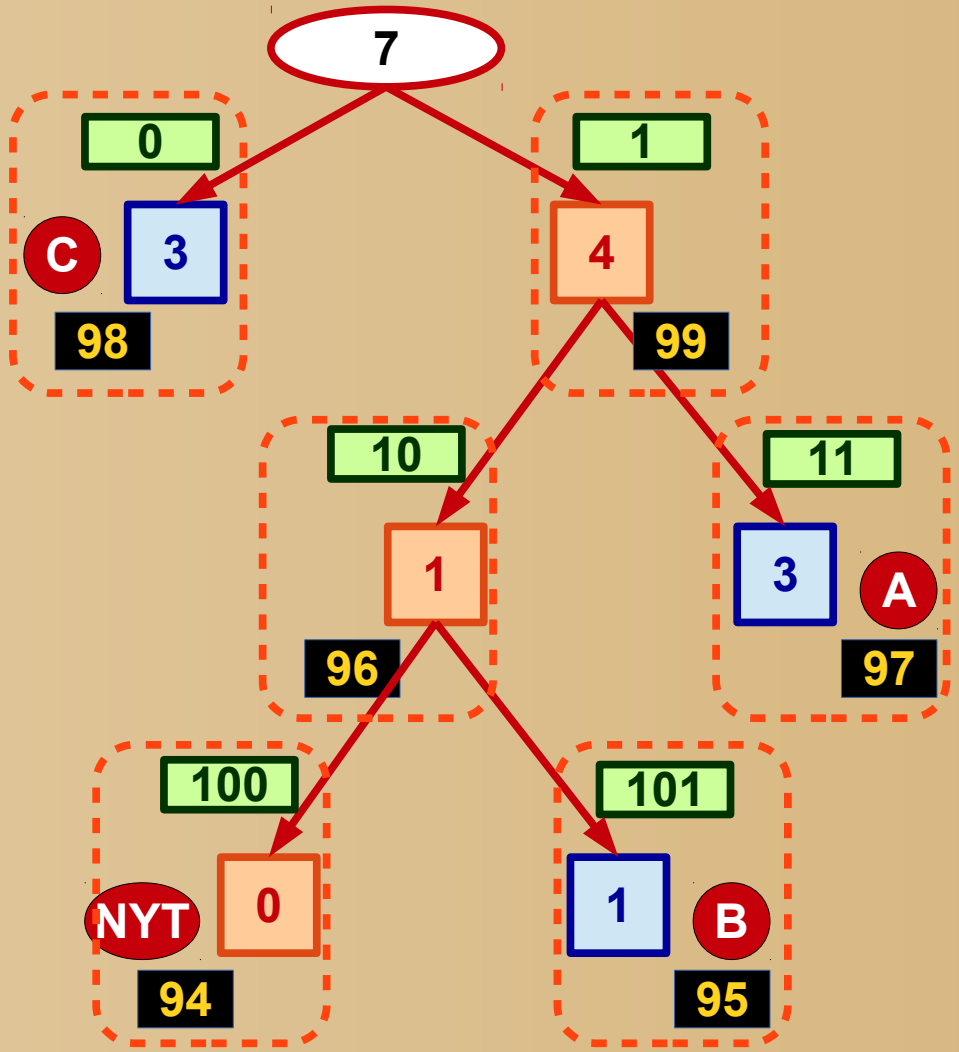
Example: Adaptive Huffman De-Compression

جامعة القاهرة
كلية الحاسبات و المعلومات

0 0 0 0 1 0 0 1 0 1 0 1 0 0 0 1 0 1 1 1 0

A nyt B nyt C C C A A

Symbol	Short Code
A	00
B	01
C	10



ABCCCAA

Parse Code

Update Tree

Swap: Done [98] , [99]

Inc Counter: Node 99 {3+}

Goto Parent: Root,

Inc {6+}

Prof. Khaled Mostafa
khaledms@fci-cu.edu.eg

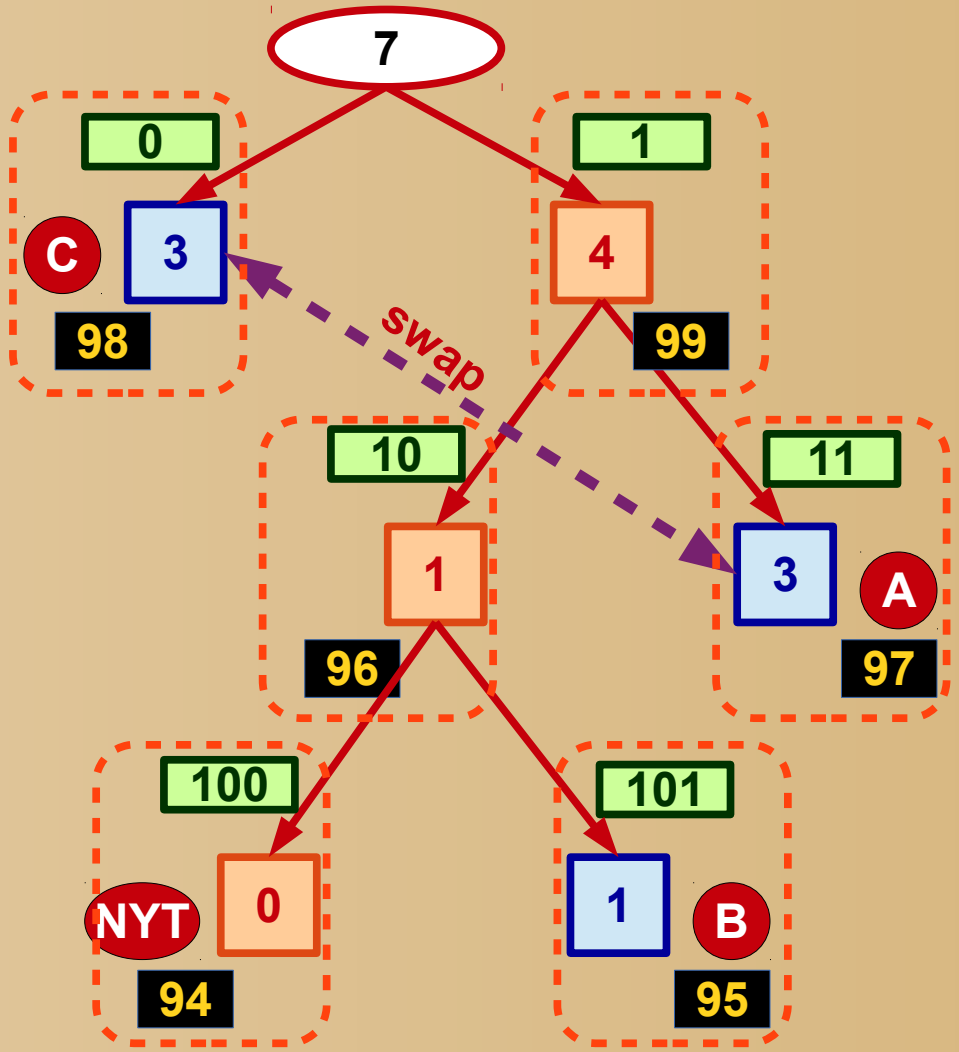
Example: Adaptive Huffman De-Compression

جامعة القاهرة
كلية الحاسبات و المعلومات

0 0 0 0 1 0 0 1 0 1 0 1 0 0 0 1 0 1 1 1 0

A nyt B nyt C C C A A A

Symbol	Short Code
A	00
B	01
C	10



ABCCCAAA

Parse Code

Code = [11] → "A"

Update Tree

Go To Node: 97

Need Swap: YES [97] , [98]

Prof. Khaled Mostafa
khaledms@fci-cu.edu.eg

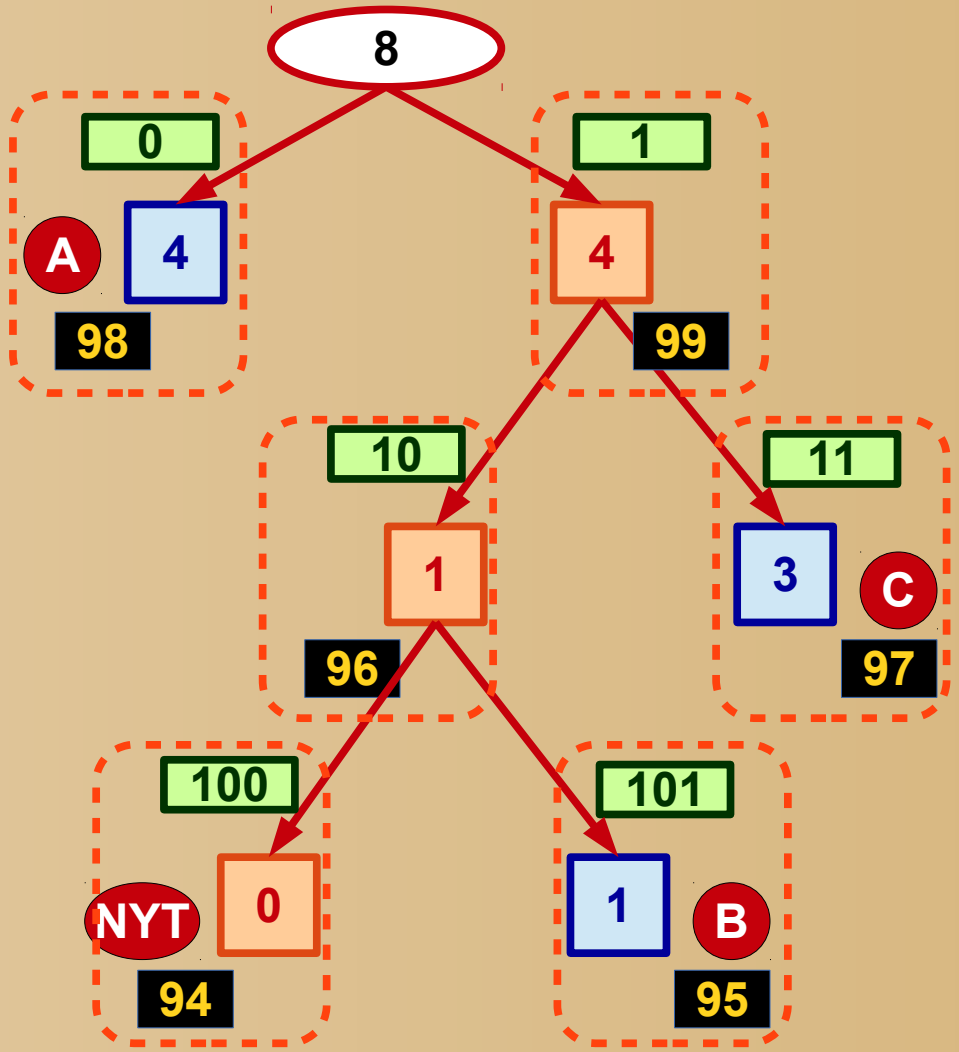
Example: Adaptive Huffman De-Compression

جامعة القاهرة
كلية الحاسبات و المعلومات

0 0 0 0 1 0 0 1 0 1 0 1 0 0 0 1 0 1 1 1 0

A nyt B nyt C C C A A A

Symbol	Short Code
A	00
B	01
C	10



ABCCCAAA

Parse Code

Update Tree

Swap: Done [98] , [97]

Inc Counter: Node 98 {3+}

Goto Parent: Root,

Inc {7+}

Prof. Khaled Mostafa
khaledms@fci-cu.edu.eg

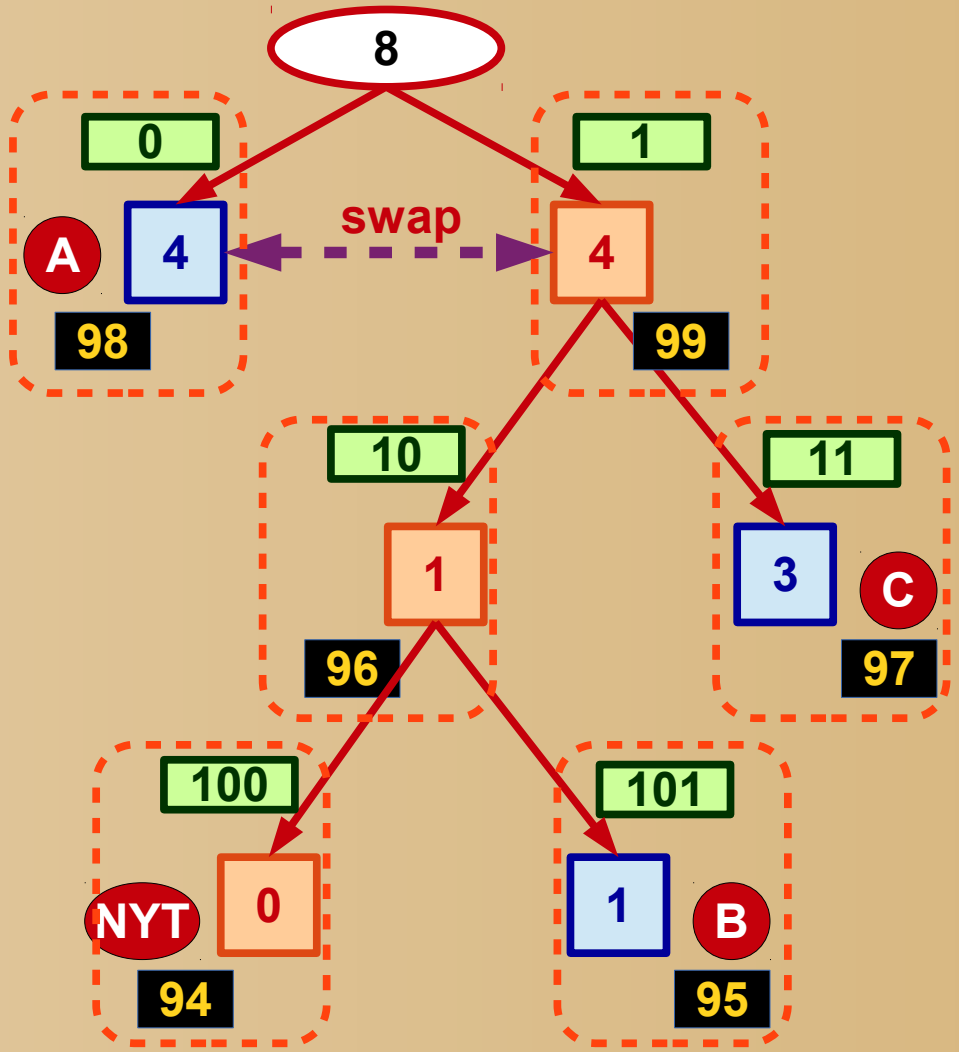
Example: Adaptive Huffman De-Compression

جامعة القاهرة
كلية الحاسبات و المعلومات

0 0 0 0 1 0 0 1 0 1 0 1 0 0 0 1 0 1 1 1 0

A nyt B nyt C C C A A A A

Symbol	Short Code
A	00
B	01
C	10



ABCCCAAAA

Parse Code

Code = [0] → "A"

Update Tree

Go To Node: 98

Need Swap: YES [98] , [99]

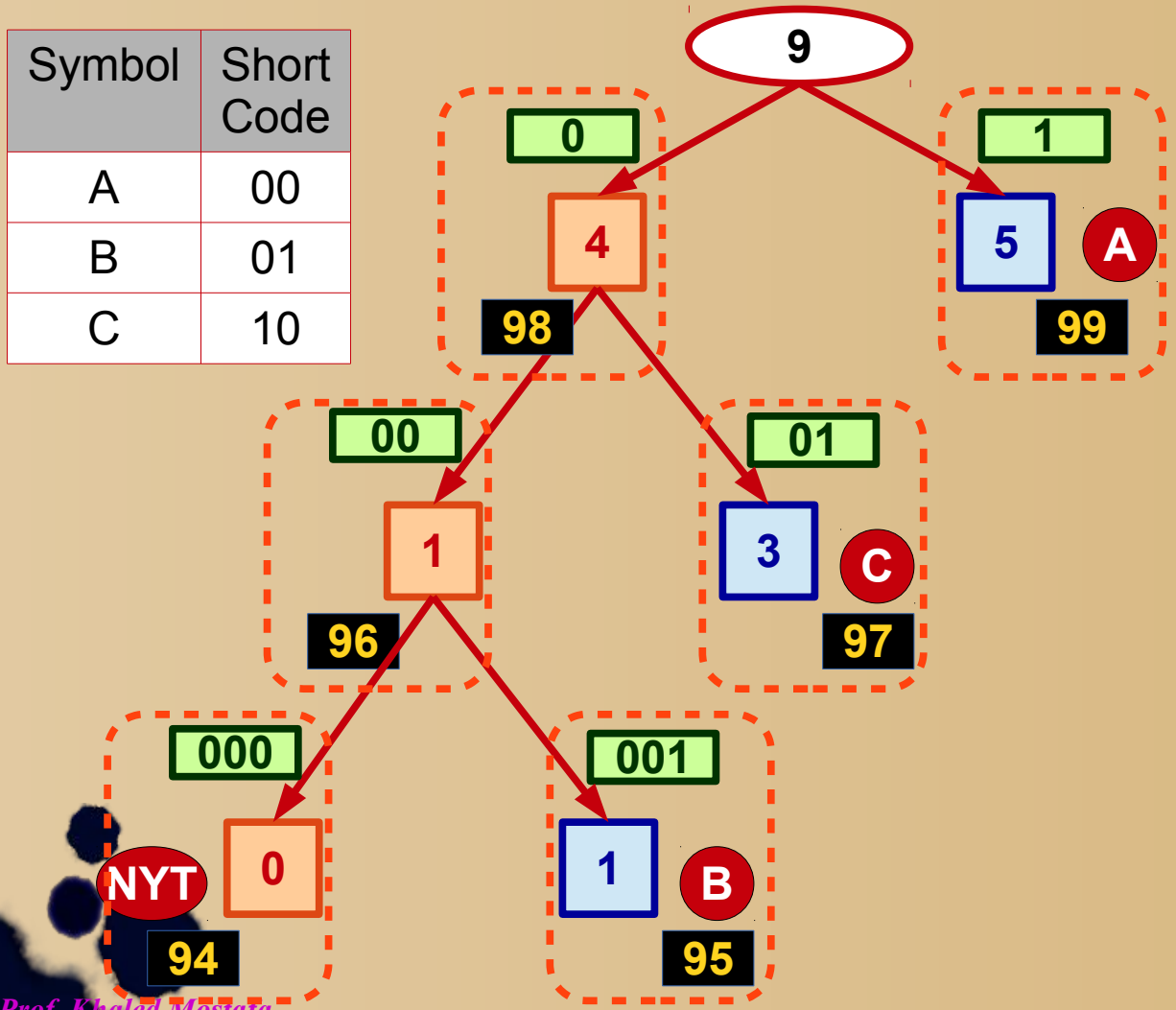
Prof. Khaled Mostafa
khaledms@fci-cu.edu.eg

Example: Adaptive Huffman De-Compression

0 0 0 0 1 0 0 1 0 1 0 1 0 0 0 1 0 1 1 1 0

A nyt B nyt C C C A A A A

Symbol	Short Code
A	00
B	01
C	10



ABCCCAAAA

Parse Code

Update Tree

Swap: Done [98] , [99]

Inc Counter: Node 99 {5+}

Goto Parent: Root,

Inc {8+}

Prof. Khaled Mostafa
khaledms@fci-cu.edu.eg

جامعة القاهرة
كلية الحاسبات
و المعلومات

Not Used Slides

Prof. Khaled Mostafa
khaledms@fci-cu.edu.eg

7A.1 Introduction

- Compression is achieved by removing data redundancy while preserving information content.
- The information content of a group of bytes (a message) is its *entropy*.
  - Data with low entropy permit a larger compression ratio than data with high entropy.
- Entropy, *H*, is a function of symbol frequency. It is the weighted average of the number of bits required to encode the symbols of a message:

$$H = -P(x) \times \log_2 P(x_i)$$

7A.1 Introduction

- The entropy of the entire message is the sum of the individual symbol entropies.

$$\sum -P(x) \times \log_2 P(x_i)$$

- The average redundancy for each character in a message of length l is given by:

$$\sum P(x) \times l_i - \sum -P(x) \times \log_2 P(x_i)$$

7A.2 Statistical Coding

- Consider the message: **`HELLO WORLD!`**
  - The letter `L` has a probability of 3/12 = 1/4 of appearing in this message. The number of bits required to encode this symbol is $-\log_2(1/4) = 2$.
- Using our formula, $\sum -P(x) \times \log_2 P(x_i)$, the average entropy of the entire message is 3.022.
  - This means that the theoretical minimum number of bits per character is 3.022.
- Theoretically, the message could be sent using only 37 bits. ($3.022 \times 12 = 36.26$)