

CS 1501

Algorithm Implementation

Recitation 9

Fall '14 – Mar. 17th

TA

Adriano Maron

University of Pittsburgh



University of Pittsburgh

DEPARTMENT OF
COMPUTER SCIENCE



Ex. 1

Create a non-binary Huffman tree of radix $r = 4$ where the frequencies of the characters are shown below:

a: 0.2
b: 0.2
c: 0.15
d: 0.15
e: 0.15
f: 0.1
g: 0.025
h: 0.025

Determine the codewords, a sequence of base- r digits and compute μ , the average number of base-4 digits per character. What is the pre-order representation of this non-binary Huffman tree?



Ex. 1

Create a non-binary Huffman tree of radix $r = 4$ where the frequencies of the characters are shown below:

a: 0.2
b: 0.2
c: 0.15
d: 0.15
e: 0.15
f: 0.1
g: 0.025
h: 0.025

Determine the codewords, a sequence of base- r digits and compute μ , the average number of base-4 digits per character. What is the pre-order representation of this non-binary Huffman tree?

1 - Determine the number of dummies required.

$$\# \text{ dummies} = \# \text{ characters} - (r + a*(r - 1)) = 4 + 3*a$$

This implies $3*a = 4$ or $a = 4/3$ (however this must be an integer)

$$\text{So, } a = \text{ceiling}(4/3) = 2.$$

$$\begin{aligned} \# \text{ dummies} &= |8 - (4 + a*(4 - 1))| \\ &= |8 - (4 + 2*3)| \\ &= 2 \end{aligned}$$



Ex. 1

Create a non-binary Huffman tree of radix $r = 4$ where the frequencies of the characters are shown below:

a: 0.2

b: 0.2

c: 0.15

d: 0.15

e: 0.15

f: 0.1

g: 0.025

h: 0.025

d1: 0

d2: 0

Determine the codewords, a sequence of base- r digits and compute μ , the average number of base-4 digits per character. What is the pre-order representation of this non-binary Huffman tree?

1 - Determine the number of dummies required.

$$\# \text{ dummies} = \# \text{ characters} - (r + a*(r - 1)) = 4 + 3*a$$

This implies $3*a = 4$ or $a = 4/3$ (however this must be an integer)

$$\text{So, } a = \text{ceiling}(4/3) = 2.$$

$$\begin{aligned} \# \text{ dummies} &= |8 - (4 + a*(4 - 1))| \\ &= |8 - (4 + 2*3)| \\ &= 2 \end{aligned}$$

2 new dummy nodes



Ex. 1

Create a non-binary Huffman tree of radix $r = 4$ where the frequencies of the characters are shown below:

a: 0.2

b: 0.2

c: 0.15

d: 0.15

e: 0.15

f: 0.1

g: 0.025

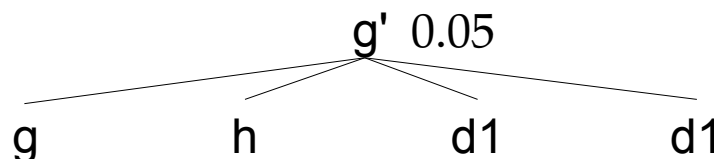
h: 0.025

d1: 0

d2: 0

Determine the codewords, a sequence of base- r digits and compute μ , the average number of base-4 digits per character. What is the pre-order representation of this non-binary Huffman tree?

2 - Form the first tree by combining the four smallest frequency characters together



g'



Ex. 1

Create a non-binary Huffman tree of radix $r = 4$ where the frequencies of the characters are shown below:

a: 0.2

b: 0.2

c: 0.15

d: 0.15

e: 0.15

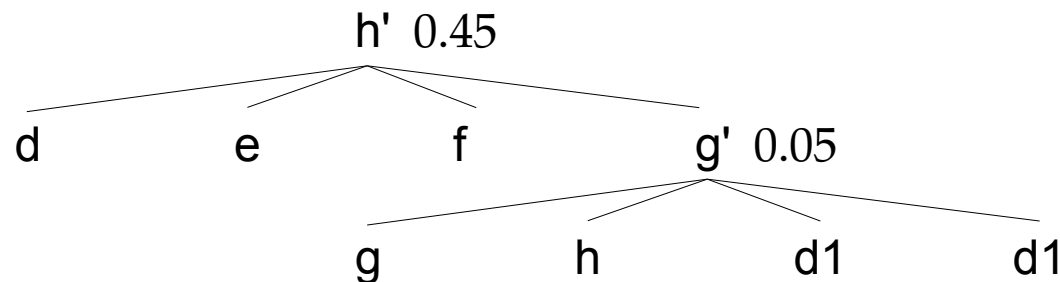
f: 0.1

g': 0.05

Determine the codewords, a sequence of base- r digits and compute μ , the average number of base-4 digits per character. What is the pre-order representation of this non-binary Huffman tree?

3 - Keep combining the four smallest frequency characters/trees together

h'





Ex. 1

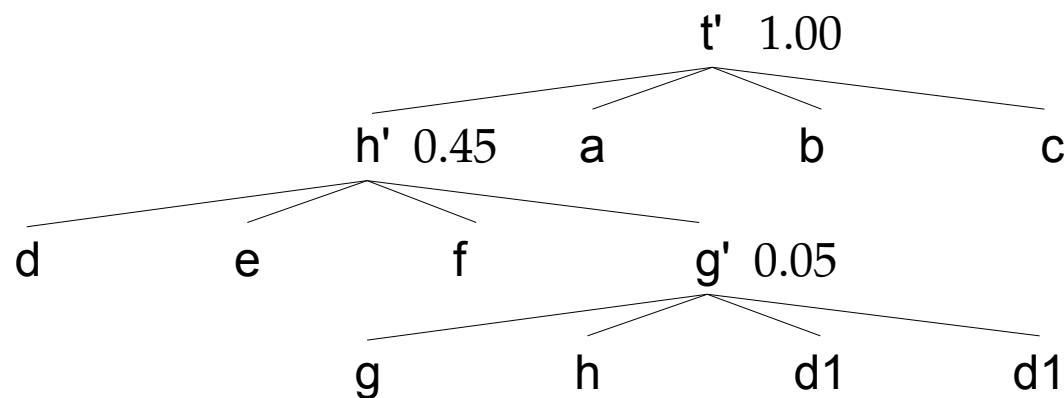
Create a non-binary Huffman tree of radix $r = 4$ where the frequencies of the characters are shown below:

h': 0.45	
a: 0.2	
b: 0.2	
c: 0.15	

t'

Determine the codewords, a sequence of base- r digits and compute μ , the average number of base-4 digits per character. What is the pre-order representation of this non-binary Huffman tree?

3 - Keep combining the four smallest frequency characters/trees together





Ex. 1

Create a non-binary Huffman tree of radix $r = 4$ where the frequencies of the characters are shown below:

Determine the codewords, a sequence of base- r digits and compute μ , the average number of base-4 digits per character. What is the pre-order representation of this non-binary Huffman tree?

4 – What are the codewords?

a: 1

b: 2

c: 3

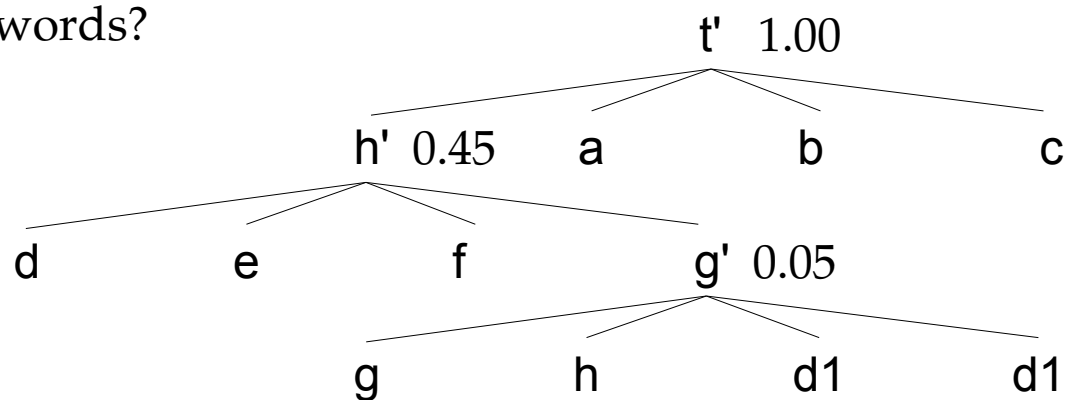
d: 00

e: 01

f: 02

g: 030

h: 031





Ex. 1

Create a non-binary Huffman tree of radix $r = 4$ where the frequencies of the characters are shown below:

Determine the codewords, a sequence of base- r digits and compute μ , the average number of base-4 digits per character. What is the pre-order representation of this non-binary Huffman tree?

a: 0.2

b: 0.2

c: 0.15

d: 0.15

e: 0.15

f: 0.1

g: 0.025

h: 0.025

5 – What is the average number of digits per character?

a: 1

Ignore dummies d1 and d2!

b: 2

$$\mu = 1 \cdot .2 + 1 \cdot .2 + 1 \cdot .15 + 2 \cdot .15 + 2 \cdot .15 + 2 \cdot .1 + 3 \cdot .025 + 3 \cdot .025 = 1.5$$

c: 3

1.5 base-4 digit per character

d: 00

e: 01

f: 02

g: 030

h: 031



Ex. 1

Create a non-binary Huffman tree of radix $r = 4$ where the frequencies of the characters are shown below:

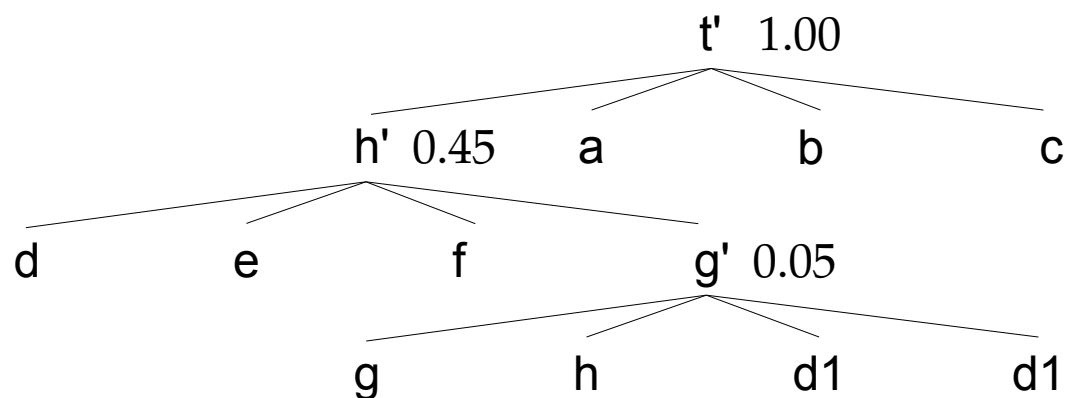
Determine the codewords, a sequence of base- r digits and compute μ , the average number of base-4 digits per character. What is the pre-order representation of this non-binary Huffman tree?

5 – What is the pre-order representation of the tree?

The pre-order representation of this tree is: 4 **def*gh\$\$abc

(get the pre-order by thinking about a DFS in the tree)

(Note: * represents an internal node and \$ represents a dummy node.)





Ex. 2

Adaptive Huffman uses a scheme to make the ASCII characters (8-bits) received have a much shorter binary representation to make the compression greater.

Suppose the only the characters transmitted are the letters A-Z and a-z (52 different characters).

Use $52 = 2^e + r$ where $0 \leq r < 2^e$, find e and r and figure out the bit patterns for each character below.

Remember, character s_k is represented as an $(e+1)$ -bit binary number for $(k-1)$ when $0 \leq k \leq 2^e$ and s_k is represented as an e -bit binary number for $(k - 2^e - 1)$ when $k > 2^e$.

$s_1 = 'A'$ $s_2 = 'B'$... $s_{26} = 'Z'$ $s_{27} = 'a'$ $s_{28} = 'b'$.. $s_{40} = 'n'$... $s_{52} = 'z'$



Ex. 2

Adaptive Huffman uses a scheme to make the ASCII characters (8-bits) received have a much shorter binary representation to make the compression greater.

Suppose the only the characters transmitted are the letters A-Z and a-z (52 different characters).

Use $52 = 2^e + r$ where $0 \leq r < 2^e$, find e and r and figure out the bit patterns for each character below.

$52 = 2^e + r$	e is small	r is large
$52 = 2^5 + 20$	$e = 5$	$r = 20$



Ex. 2

Adaptive Huffman uses a scheme to make the ASCII characters (8-bits) received have a much shorter binary representation to make the compression greater.

Suppose the only the characters transmitted are the letters A-Z and a-z (52 different characters).

Use $52 = 2^e + r$ where $0 \leq r < 2^e$, find e and r and figure out the bit patterns for each character below.

Remember, character s_k is represented as an $(e+1)$ -bit binary number for $(k-1)$ when $0 \leq k \leq 2^*r$ and s_k is represented as an e -bit binary number for $(k - r - 1)$ when $k > 2^*r$.

How to represent each s_k

$$e = 5$$

$$r = 20$$

If $0 \leq k \leq 2^*r$

$$s_k = \text{intToBin}(k-1, e+1)$$

Else

$$s_k = \text{intToBin}(k-r-1, e)$$



Ex. 2

$e = 5$
 $r = 20$

How to represent each s_k

If $0 \leq k \leq 2 \cdot r$

$s_k = \text{intToBin}(k-1, e+1)$

Else

$s_k = \text{intToBin}(k-r-1, e)$

$b = \text{intToBin}(i, nB)$

i : input integer

nB : number of bits
in b

s1	s2	...	s26	s27	s28	...	s40	...	s52
A	B	...	Z	a	b	...	h	...	z
000000	000001	...	011001	011010	011011	...	100111	...	11111



Ex. 1

Draw the trees (number the nodes) created by the UPDATE procedure (Adaptive Huffman) after each of these characters are received: b, c, b, a, a, b, w. Assume there are $m = 26$ symbols in the alphabet.



Ex. 1

Draw the trees (number the nodes) created by the UPDATE procedure (Adaptive Huffman) after each of these characters are received: b, c, b, a, a, b, w. Assume there are $m = 26$ symbols in the alphabet.

NYT (Not Yet Transmitted) = $2m - 1 = 51$

0 NYT
51

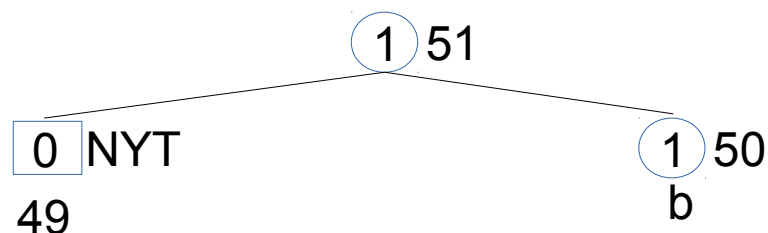


Ex. 1

Draw the trees (number the nodes) created by the UPDATE procedure (Adaptive Huffman) after each of these characters are received: b, c, b, a, a, b, w. Assume there are $m = 26$ symbols in the alphabet.

NYT (Not Yet Transmitted) = $2m - 1 = 51$

Receiving b



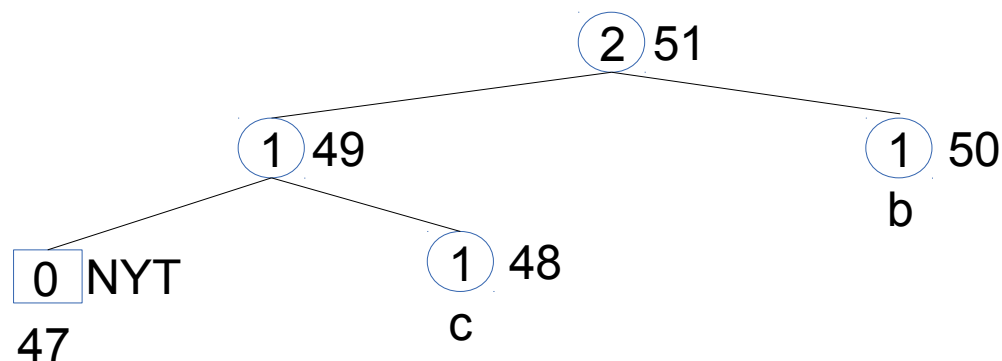


Ex. 1

Draw the trees (number the nodes) created by the UPDATE procedure (Adaptive Huffman) after each of these characters are received: b, c, b, a, a, b, w. Assume there are $m = 26$ symbols in the alphabet.

NYT (Not Yet Transmitted) = $2m - 1 = 51$

Receiving c



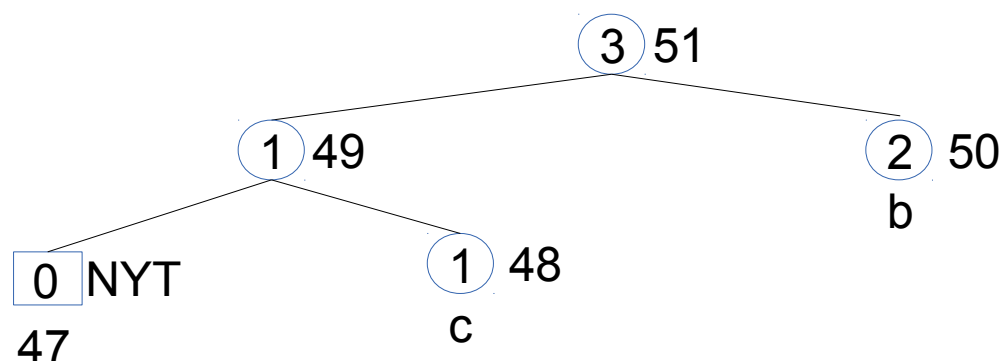


Ex. 1

Draw the trees (number the nodes) created by the UPDATE procedure (Adaptive Huffman) after each of these characters are received: b, c, b, a, a, b, w. Assume there are $m = 26$ symbols in the alphabet.

NYT (Not Yet Transmitted) = $2m - 1 = 51$

Receiving b



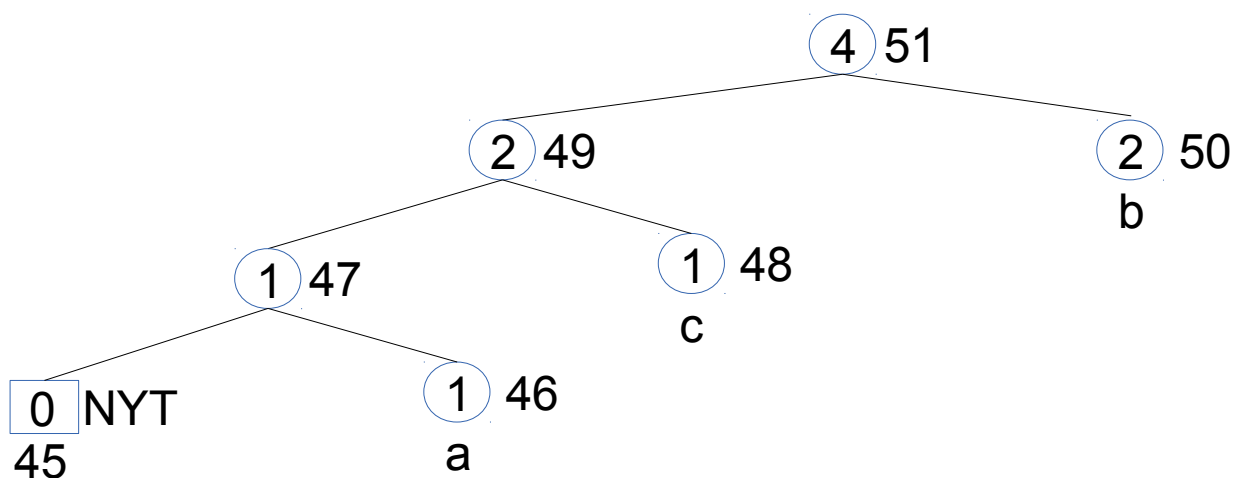


Ex. 1

Draw the trees (number the nodes) created by the UPDATE procedure (Adaptive Huffman) after each of these characters are received: b, c, b, a, a, b, w. Assume there are $m = 26$ symbols in the alphabet.

NYT (Not Yet Transmitted) = $2m - 1 = 51$

Receiving a



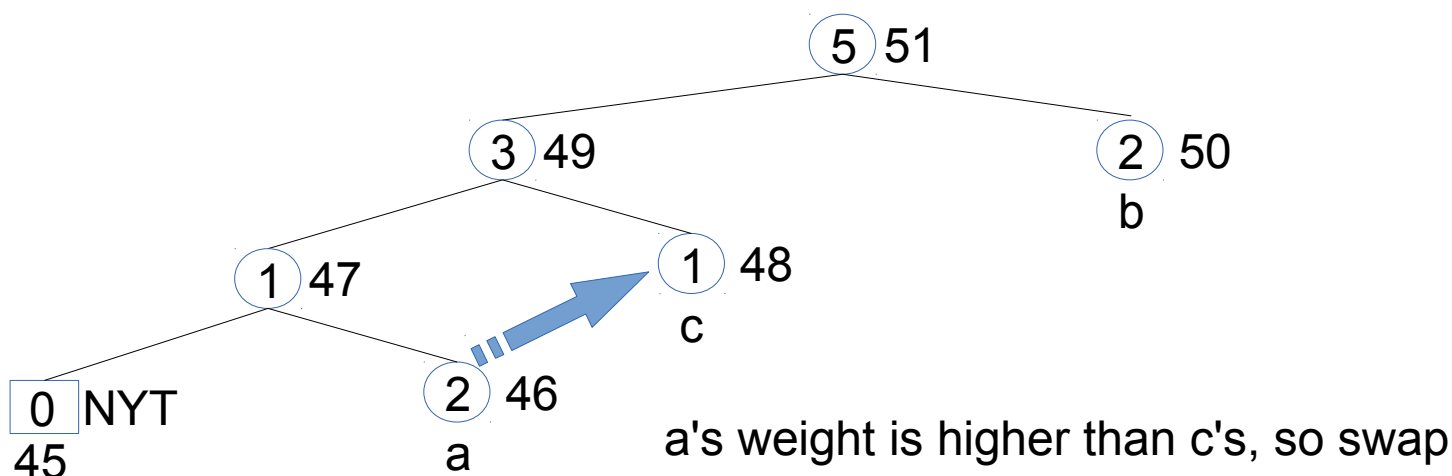


Ex. 1

Draw the trees (number the nodes) created by the UPDATE procedure (Adaptive Huffman) after each of these characters are received: b, c, b, a, a, b, w. Assume there are $m = 26$ symbols in the alphabet.

NYT (Not Yet Transmitted) = $2m - 1 = 51$

Receiving a



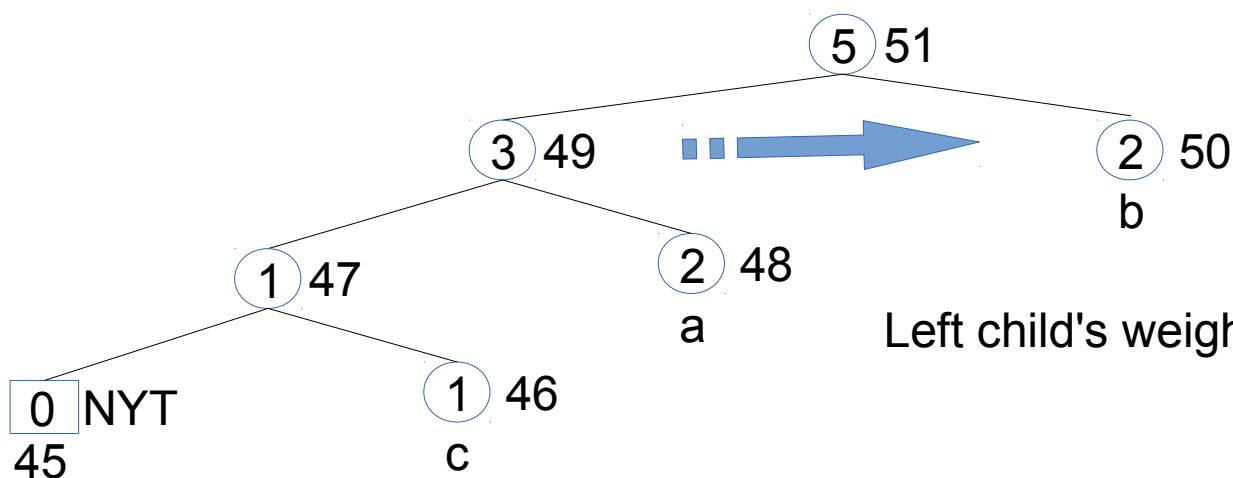


Ex. 1

Draw the trees (number the nodes) created by the UPDATE procedure (Adaptive Huffman) after each of these characters are received: b, c, b, a, a, b, w. Assume there are $m = 26$ symbols in the alphabet.

NYT (Not Yet Transmitted) = $2m - 1 = 51$

Receiving a



Left child's weight is higher than right child's, so swap

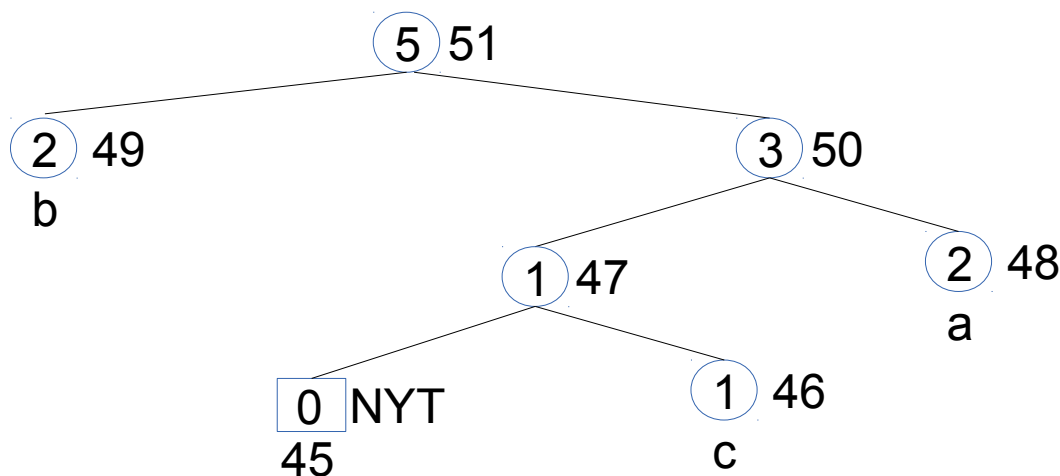


Ex. 1

Draw the trees (number the nodes) created by the UPDATE procedure (Adaptive Huffman) after each of these characters are received: b, c, b, a, a, b, w. Assume there are $m = 26$ symbols in the alphabet.

NYT (Not Yet Transmitted) = $2m - 1 = 51$

Receiving a





Ex. 1

Draw the trees (number the nodes) created by the UPDATE procedure (Adaptive Huffman) after each of these characters are received: b, c, b, a, a, b, w. Assume there are $m = 26$ symbols in the alphabet.

NYT (Not Yet Transmitted) = $2m - 1 = 51$

Receiving w

