9 Write down the algorithmic steps for Shanno-Fano Encoding.

=> Algorithm:

1 Find out the Probability For each code.

2 The message or code return in the order of decreasing Probability.

3 then divided message into two most equal subset (X & Y).

4 The message of set X is give 0 bit and the message with Y is give 1 bit.

5 Applied this step for each step until the end.

6 After Gatting the all the symbol codeword calculate each symbol entropy.

$$H = -\sum_{i=1}^{n} P_i \log_2 P_i$$

SMVS
Page No.
Date : / /

7 After the entropy, calculate the efficiency,

$$\eta = \frac{H}{\hat{H}}$$

where $\hat{H} = \sum_{i=1}^{n} P_i n_i$

8 Calculate Redundancy $R_e = 1 - \eta$

8 Explain Huffman Algorithm with example.

=> The Huffman Algorithm is a greedy algorithm used for lossless data compression.

The Huffman coding is to assign ~~vara~~ variable-length codes to input characters.

In Huffman Algorithm, shorter codes assigned to more frequent characters and longer code assigned to less frequent characters

SMVS
Page No.
Date : / /

=> Algorithm :

1 Create a sorted node based on Probability and arrange it in descending order.

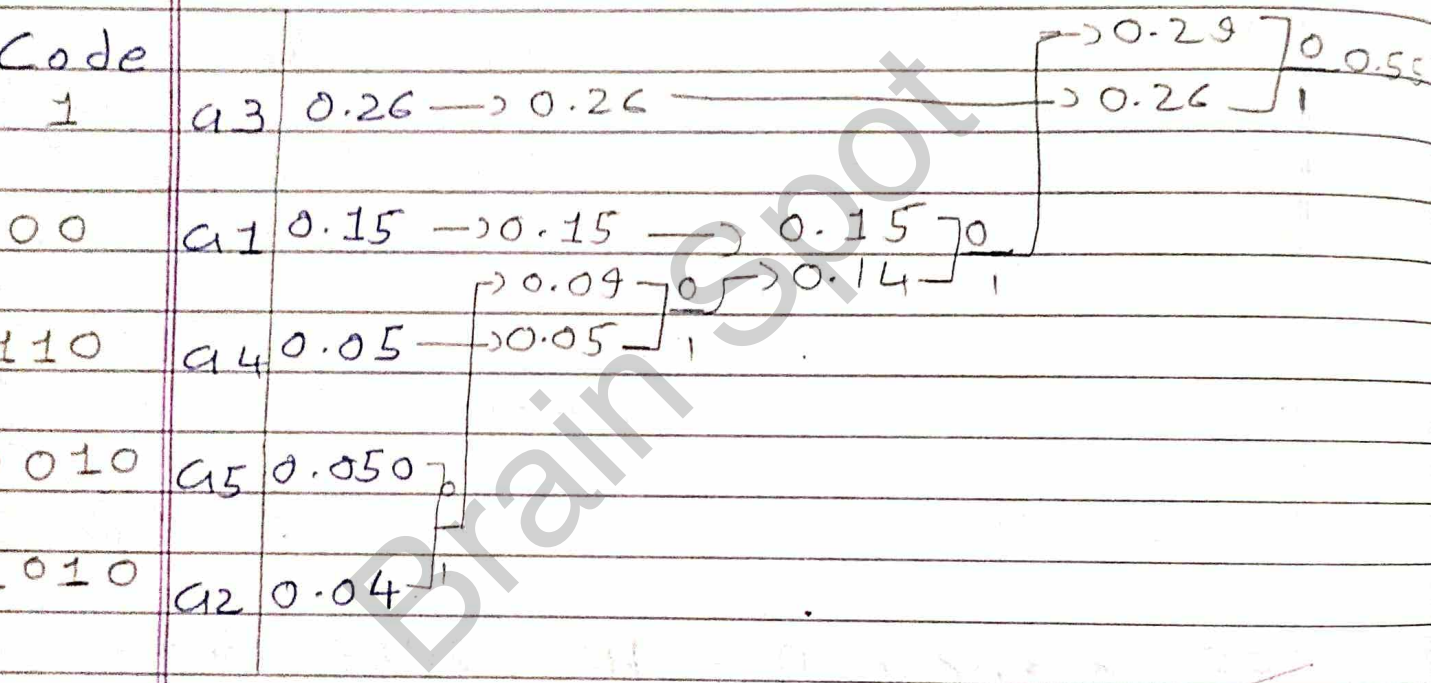
2 Repeat next step till all data complete.

3 Take a smallest node do the sumation and remove that node.

4 Based on sumation on above step create a new node.

SMVS
Page No.
Date : / /

4 Consider source alphabet of A = {a1, a2, a3, a4, a5} with having probability, P(a1) = 0.15, P(a2) = 0.04, P(a3) = 0.26, P(a4) = 0.05, P(a5) = 0.050. Calculate entropy, redundancy using Huffman code.



$$H(X) = -\sum_{i=1}^{n} P_i \log_2 P_i$$

$$= -(0.26 \times \log_2 0.26 + 0.15 \log_2 0.15 + 0.05 \log_2 0.05 + 0.05 \log_2 0.05 + 0.04 \log_2 0.04)$$

$$= 0.505 + 0.41 + 0.432 + 0.185$$

$$= 1.532$$

SMVS
Page No.
Date : / /

$$\hat{H} = \sum_{i=1}^{n} P_i n_i$$

$= (0.26 \times 1 + 0.15 \times 2 + 0.05 \times 3 + 0.05 \times 4 + 0.04 \times 4)$

$= 1.07$

Efficiency $\eta = \frac{H}{\hat{H}} = \frac{1.532}{1.07} = 1.43$

Redundancy $= 1 - \eta = 1 - 1.43 = 0.43$

Average Length of Code = 1.07

1 Consider a source emits after letter from a alphbet A = {a1, a2, a3, a4} with probability, P(a1) = 0.2, P(a2) = 0.4, P(a3) = 0.2, P(a4) = 0.1 and P(a5) = 0.1. Find Huffman code and Average length. Also Find shanno Fano code for the same.

=> Shanno-Fanon :

	Pro.					Code	L_Co	H(x)
a2	0.4	0				0	1	-0.53
a1	0.2	1	0			10	2	-0.46
a3	0.2	1	1	0		110	3	-0.46
a4	0.1	1	1	1	0	111	3	-0.33
a5	0.1	1	1	1	1	1111	4	-0.33

$$H = -\sum_{i=1}^{n} P_i \log_2 P_i$$

$$= -(0.4 \log_2 0.4 + 0.2 \log_2 0.2 + 0.2 \log_2 0.2 + 0.1 \log_2 0.1 + 0.1 \log_2 0.1)$$

$$= 2.11$$

$$\hat{H} = \sum_{i=1}^{n} P_i n_i$$

$$= (0.4 \times 1 + 0.2 \times 2 + 0.2 \times 3 + 0.1 \times 3 + 0.1 \times 4)$$

$$= 2.1$$

$$\text{Efficiency } \eta = \frac{H}{\hat{H}} = \frac{2.11}{2.1} = 1$$

$$\text{Redundancy } R_\theta = 1 - \eta$$

$$= 1 - 1$$

$$= 0$$

10 List out the application of Huffman code and explain in details.

=> This are the basic application of Huffman code.

1 File Compression: Huffman coding is helps to reduce the size of the file by representing frequently occuring character with shorter codes.

2 Image Compression: Huffman coding helps to represent image pixel values with

variable-length codes.

3 Text Compression : Huffman Coding is used in text compression algorithm like Burrows-Wheeler Transform or Move-to-Front etc.

4 Network Communication : Huffman coding is used in network communication protocols to compress data before transmission.

5 Data Storage : Huffman coding is applied in various data storage system to compress data before storing in storage media.

6 Multimedia Compression : Huffman coding is used to reduce the size of multimedia file, enabling efficient storage.

7 Encryption : Huffman Coding is utilized in some encryption algorithm to encode specific pattern of data.

11 Explain update procedure for Adaptive Huffman coding. Also explain Encoding and Decoding.

=> Adaptive Huffman Coding can be Perform using three steps.

(1) Update Procedure
(2) Encoding
(3) Decoding.

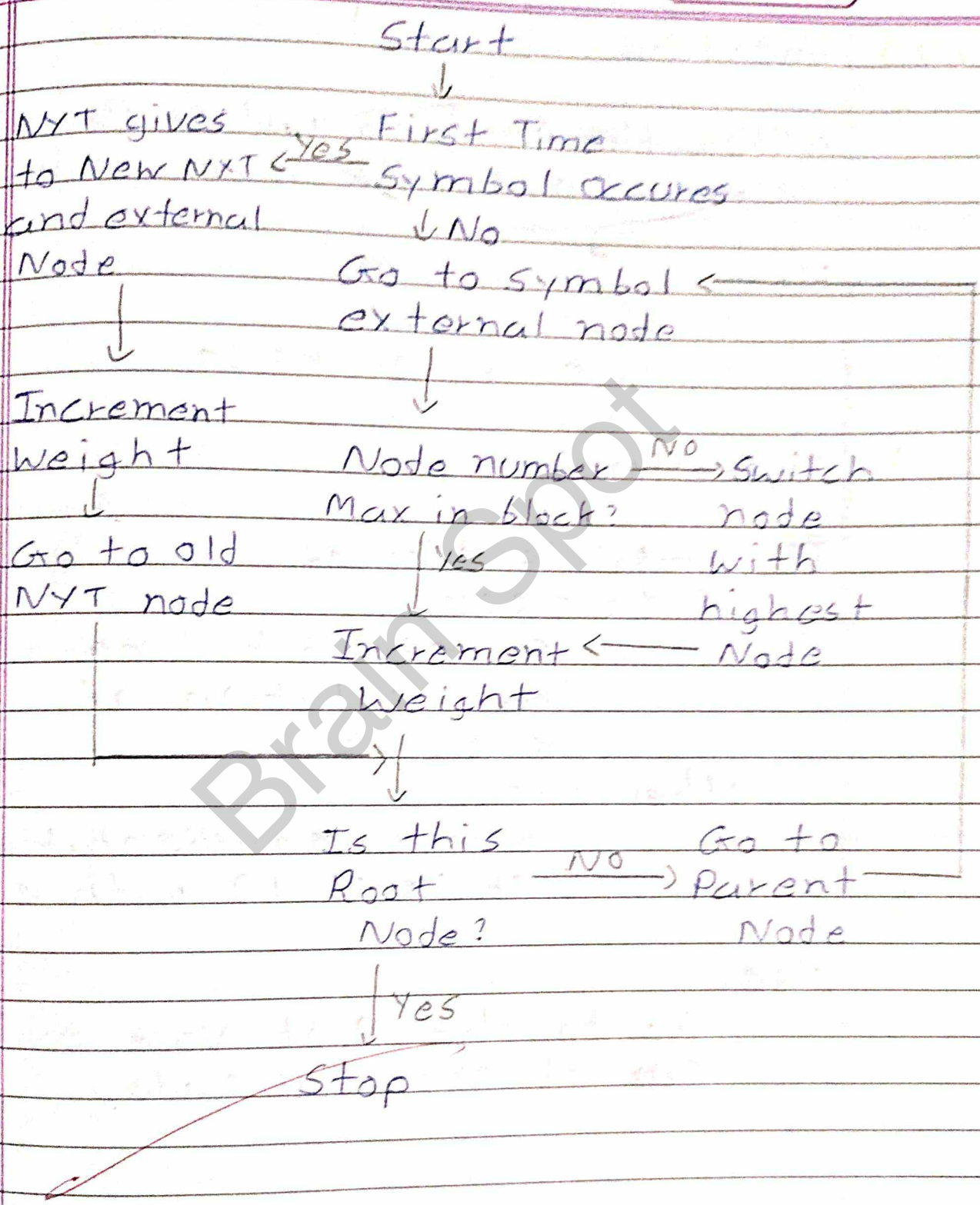
(1) Update Procedure:

In Adaptive Huffman Coding, Tree we have to take NYT Node as a Right child of every tree node.

Always value of NYT is zero.

Weights = (2 * n) - 1
= 2 * 26 - 1
= 51

where, n = Total alphabat Number.



(2.) Encoding:

After the Perform Update Procedure we have to encode every symbol.

Here, $2^e + r = m$

$\therefore 2^e + r = 26$

$\therefore e = 4$ or $r = 10$

→ Algorithm:

IF Symbol is occures,

if $1 \leq k \leq 2r$.

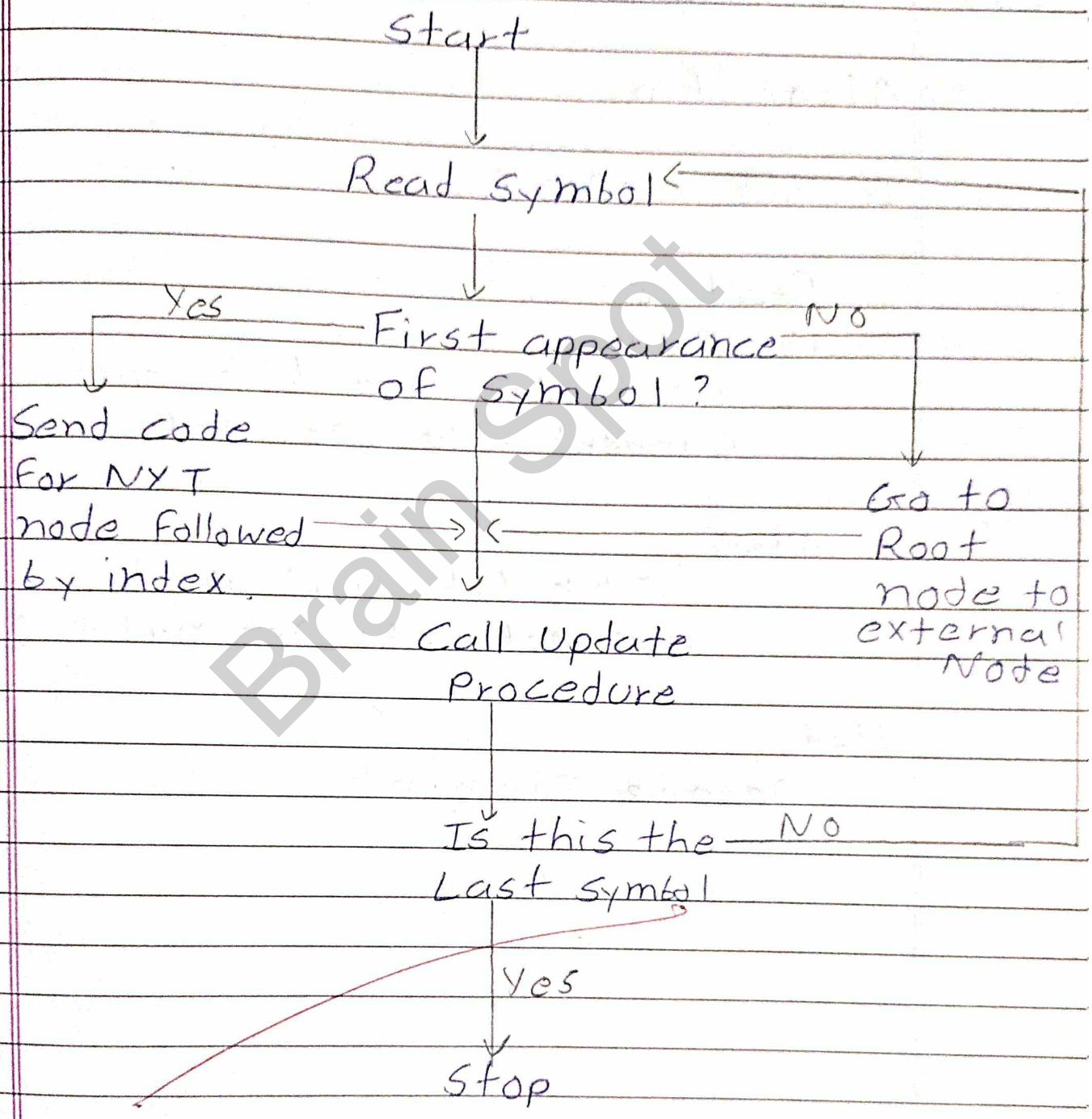
So, e+1 bit binary representation of k-1 bit.

otherwise,

binary representation of (k-r-1) with e bit

else.

Go to the Root Node to Symbol external Node.


Start
Read Symbol
First appearance of Symbol ?
Yes
No
Send code For NYT node followed by index.
Go to Root node to external Node
Call Update Procedure
Is this the Last Symbol
No
Yes
Stop

(3) Decoding:

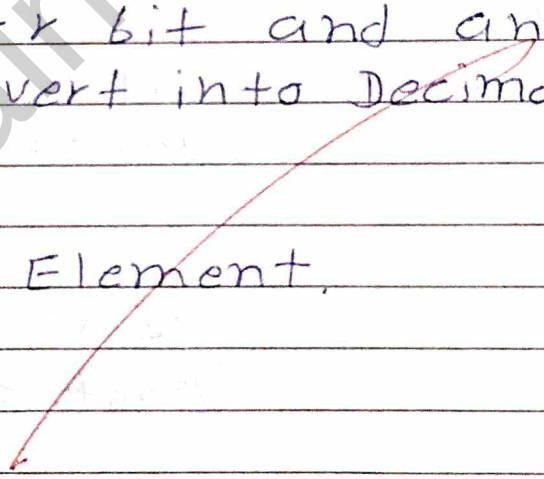
=> Algorithm:

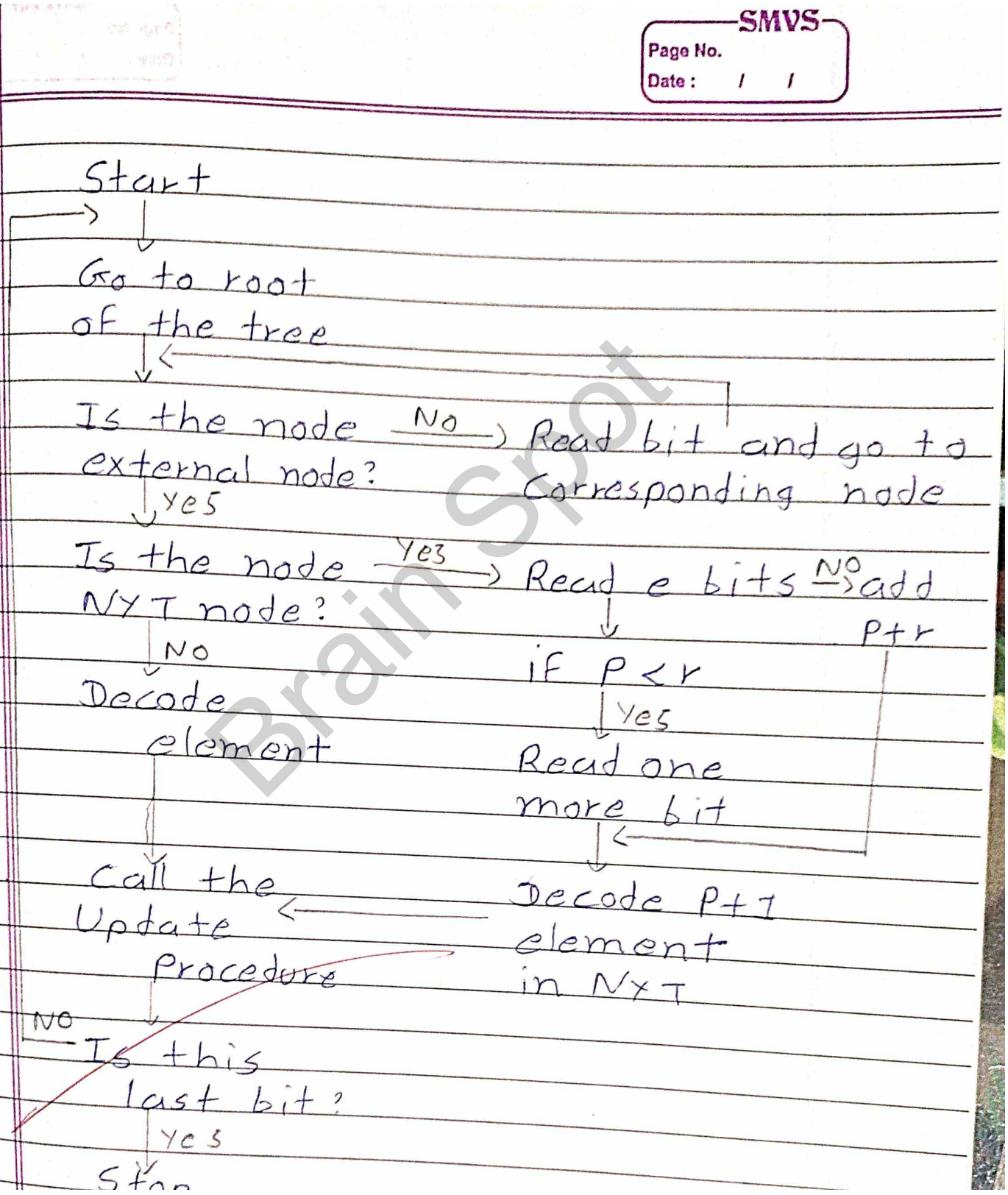
If bit is NYT

Read 'e' bit
if P < r
read 1 more bit and
Convert into Decimal

else,
P + r bit and and
Convert into Decimal + 1

else,
Decode Element.



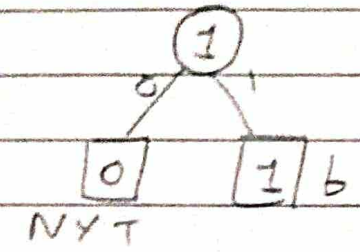

Start
Go to root of the tree
Is the node external node?
No
Read bit and go to Corresponding node
yes
Is the node NYT node?
Yes
Read e bits
No
add P+r
No
Decode element
if P<r
Yes
Read one more bit
Call the Update Procedure
Decode P+1 element in NYT
No
Is this last bit?
yes
Stop

Q6 Encode the message {BANANA} using Adaptive Huffman Coding

=>

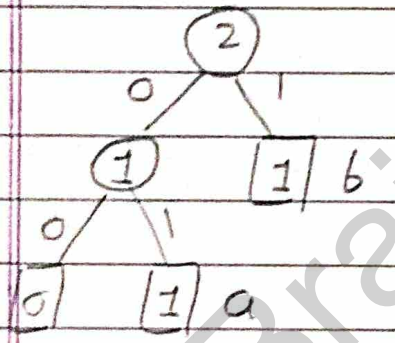
(1) [0] NYT

(2)



b = 2, 2<20, 2-1
b = 00001

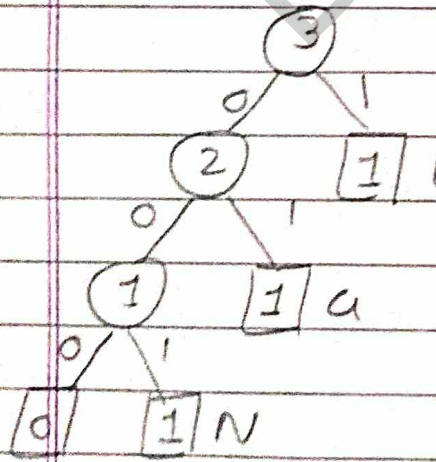
(3)



a = 1, 1<20, 1-1
NYT = 0

a = 0 00000

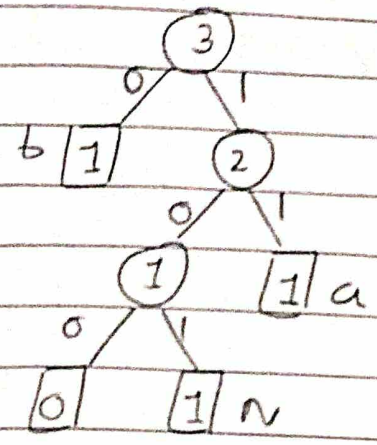
(4)



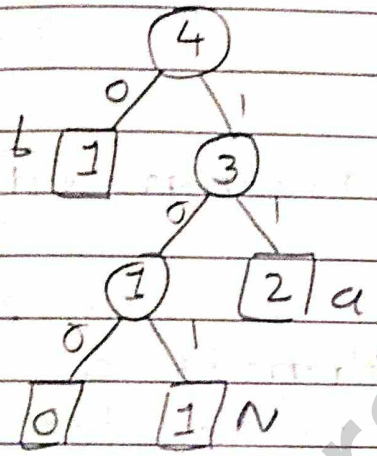
N = 14, 14<20, 14-1
NYT = 00

N = 0001101

Swap

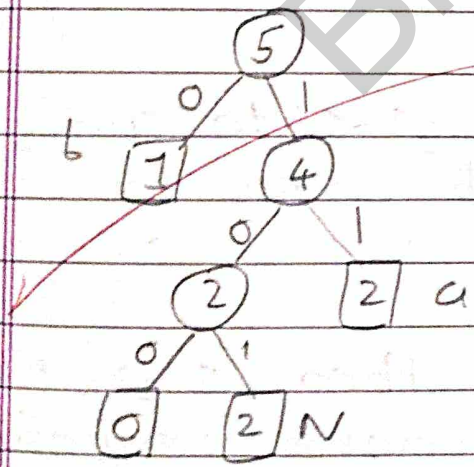


5



a Enter,
a = 11

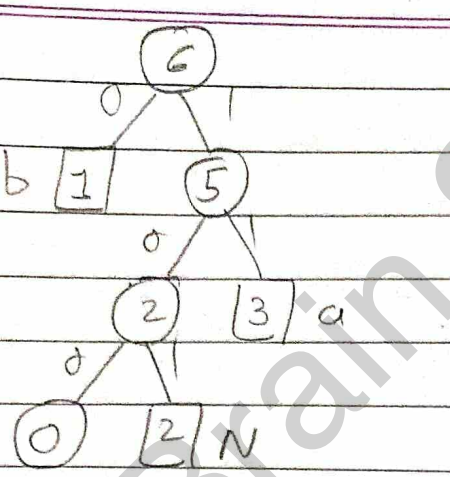
6



N Enter, N = 101

SMVS
Page No.
Date : / /

(7)



a = 11

BANANA = 0000100000 000 1101 11101
11