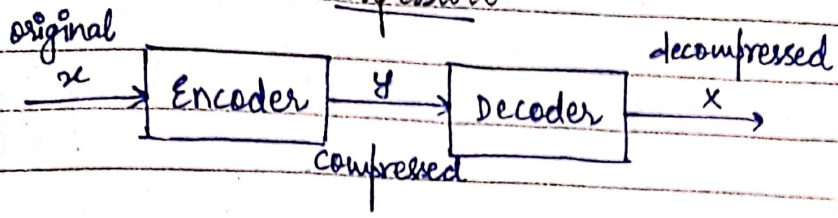
24/01/19

DATE / /201
PAGE

UNIT : 1

Compression



Classification

Compression is divided in 2 parts :-

(i) Lossless compression
- Decoded data = original data
- Compression ratio < lossy compression

Eg. LZ, JBIG

(ii) Lossy compression
- Decoded data ~ original data
- Compression ratio > lossless compression
- Used where small errors are allowed.

Eg. MPEG, JPEG.

Entropy

It's the measurement of info content for a set of message s with P(E) P(s).

Self info of s is :-

$$i(s) = \log \frac{1}{P(s)}$$

- Conditional entropy

$$H(S) = \sum_{s \in S} P(s) \log \frac{1}{P(s)}$$

Q. $P(s) = \{0.25, 0.25, 0.25, 0.125, 0.125\}$
Find $H(s)$

$$H(s) = 3\left[0.25 \log \frac{1}{0.25}\right] + 2\left[0.125 \log \frac{1}{0.125}\right]$$

$$= 3(0.25 \log 4) + 2(0.125 \log 8)$$

$$= 2.25$$

Q. $P(s) = \{0.5, 0.125, 0.125, 0.125, 0.125\}$
Find $H(s)$

$$H(s) = \left[0.5 \log \frac{1}{0.5}\right] + 4\left[0.125 \log \frac{1}{0.125}\right]$$

$$= 2$$

Q. $P(s) = \{0.75, 0.0625, 0.0625, 0.0625, 0.0625\}$
Find $H(s)$

$$H(s) = \left[0.75 \times \log \frac{1}{0.75}\right] + 4\left[0.0625 \log \frac{1}{0.0625}\right]$$

$$= \sqrt{2}$$

$$= 1.414$$

Measures of Performance

Compression algo can be evaluated in a no. of diff^{n} ways. We can measure the relative complexity of the algorithm in terms of compression ratio.

$$\text{Compression Ratio} = \frac{\text{Bits required to represent data before compression}}{\text{Bits required to represent data after compression}}$$

→

DATE / /201
PAGE

Q. Suppose storing an image of size 256×256 pixel requires 65538 bytes. after compression it requires 16384 bytes
Compression ratio = $\frac{65538}{16384}$ → 4 : 1

Shanon Fano coding

- This is a source coding method to construct binary codes
- The codes should satisfy following conditions :-

(i) No sequence of codes can be obtained from each other by adding more binary digit to shorter codes (prefix P(E))
(ii) Tx of encoded message is efficient. 1 & 0 appear ~~independtly~~ independently.

- Procedure

(i) Message are written in decreasing order of their P(E)s.
(ii) The message set is divided into 2 subsets of equal or nearly equal P(E)s
(iii) Code 0 is assign to each message in one subset above partition line.
(iv) Code 1 is assign to each message in one subset below partition line.
(v) This procedure continue until each subset contains only one message.

Q. Apply Shanon Fano coding procedure for following message
$[X] = [x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8]$
$[P] = \left[\frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{16}, \frac{1}{16}, \frac{1}{4}, \frac{1}{16}, \frac{1}{8}\right]$
take M = 2.

Message	probabilities	S_1	S_2	S_3	S_4
X_1	0.25	0	0		
X_6	0.25	0	1		
X_2	0.125	1	0	0	
X_8	0.125	1	0	1	
X_3	0.0625	1	1	0	0
X_4	0.0625	1	1	0	1
X_5	0.0625	1	1	1	0
X_7	0.0625	1	1	1	1

$$\text{Efficiency} = \eta = \frac{H(x)}{\bar{L} \cdot \log_2 M}$$

$H(x)$ - Entropy

$$\bar{L} = \sum_{1}^{n} p_k n_k$$

$$\Rightarrow \bar{L} = \frac{1}{4}\times 2 + \frac{1}{4}\times 2 + \frac{1}{8}\times 3 + \frac{1}{8}\times 3 + \frac{1}{16}\times 4 + \frac{1}{16}\times 4 + \frac{1}{16}\times 4 + \frac{1}{16}\times 4 = 2.75$$

$$H(x) = 2\left[0.25 \log \frac{1}{0.25}\right] + 2\left[0.125 \log \frac{1}{0.125}\right] + 4\left[0.0625 \log \frac{1}{0.0625}\right]$$

$$= 1 + 0.75 + 1 = 2.75$$

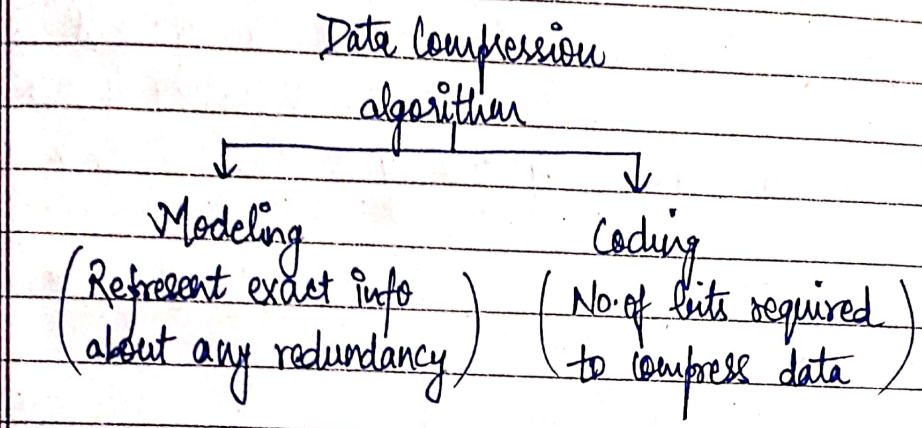
$$\text{Efficiency} = \eta = \frac{2.75}{2.75} = 1$$

27/01/19

DATE / /201
PAGE

Modelling & coding

In data compression, when we reconstruct message, the decision whether a compression scheme is to be lossy or lossless, the exact compression scheme we use will depend on a no. of diffn factors. Some of the most important factors are characteristic of data that need to be compressed



- Models

(i) Physical model

If we know something about the physics of data generation, we can use that information to construct a model.

Eg. speech related application.

Knowledge about the physics of production can be used to construct a mathematical model for a sample speech model.

(ii) Probability model

The simplest statistical model for the source is to assume that each letter that is generated by the source is independent of every other letter & each occurrence of letter have same P(E). This kind of model is known as P(E) model In this model, we consider only source.

DATE / /201
PAGE

(iii) Markov model

This is one of the most popular model. This represent dependency in data. The model is given by famous mathematician Markov. This is used in lossless compression. In this, we use discrete time frame $\{t_x\}$ & observation frame $\{X_n\}$.

The P(E) of observation :-

$$P(x_n | x_{n-1} \ldots\ldots x_{n-k}) = P(t_{x_n} | t_{x_{n-1}} \ldots\ldots t_{x_{n-k}})$$

- In other words, knowledge of past 'k' symbols is equivalent to the knowledge of entire past history of the process.
- The value taken by the set $X_{n-1} \ldots\ldots X_{n-k}$ are called the state of process.
- If the size of source alphabet is 'l' then no. of state is 'l^k'.

$$\boxed{X_n = PX_{n-1} + \epsilon_n}$$

ϵ_n → Noise

Q. Consider a binary image has a 2 type of pixel - black & white pixel. We know that the appearence of white pixel as the next observation depends on some extend whether the current pixel is white or black.

current pixel is white

S_w S_b → current pixel is black

$p(w/b)$ $p(b/w)$

$p(S_w)$ $p(S_b)$

Entropy $H = \sum_{i=1}^{M} p(S_i) H(S_i)$

$H(S_w) = -p(b/w) \log p(b/w) - p(w/b) \log p(w/b)$.

31/01/17

▶ **Coding**

It means assignment of binary sequence to element of an alphabet.

Eg.

letter	codeword
a_1	0
a_2	00
a_3	1
a_4	11

Codeword: Individual no. of code known as codeword

Alphabet: It's a collection of symbols called letters

Rate of code: The average no. of bits per symbol is called as the rate of code.

- **Uniquely decodable code**

The average length of the code isn't the only important point in designing a good code.

Q. Suppose we have 4 letters :-

a_1	$P(a_1) = 1/2$
a_2	$P(a_2) = 1/4$
a_3	$P(a_3) = 1/8$
a_4	$P(a_4) = 1/8$.

Code 1	Code 2	Code 3	Code 4
0	0	0	0
0	1	10	01
1	00	110	011
10	11	111	0111

→

- Average length (l) = $\sum p(a_i)\, n(a_i)$

$l_1 = 0.5\times1 + 0.25\times1 + 0.125\times1 + 0.125\times2$
$= 1.125$ bits/symbols

$l_2 = 0.5\times1 + 0.25\times1 + 0.125\times2 + 0.125\times2$
$= 1.25$ bits/symbols

$l_3 = 0.5\times1 + 0.25\times2 + 0.125\times3 + 0.125\times3$
$= 1.75$ bits/symbols

$l_4 = 0.5\times1 + 0.25\times2 + 0.125\times3 + 0.125\times4$
$= 1.875$ bits/symbols.

- Test for uniquie decodability

Suppose we have two binary codewords a, b. 'a' is 'k' bit long & 'b' is 'n' bit long & $k<n$. If the k bit of 'b' are identical to 'a' then 'a' is called prefix of 'b'. (n-k) bits of b be called dangling suffix.

Eg. a = 010
b = 01011
prefix, dangling suffix

Steps for finding uniquely decodable codes

(i) Construct a list of all code words.
(ii) Examine all pairs of codewords to check any codeword is prefix of another codeword.
(iii) Wherever you find such pairs, add dangling suffix to the list in previous iteration.
(iv) Now repeat the procedure with current list if we get a dangling suffix that is a codeword in the list. Then the codeword isn't uniquely decodable

DATE / /201
PAGE

Eg. $a_1 = 0$
$a_2 = 01$
$a_3 = 10$
{0, 01, 10}
{0, 01, 10, 1} Not uniquely decodable

Eg. $a_1 = 0$; $a_2 = 01$; $a_3 = 110$; $a_4 = 111$
{0, 01, 110, 111}
{0, 01, 110, 111, 1}
{0, 01, 110, 111, 1, 10} Not uniquely decodable
{0, 01, 110, 111, 1, 10, 11}

~~Prefix code~~

{0, 10, 110, 111} {0, 01, 011, 0111}
{0, 10, 110, 111} {0, 01, 011, 0111, 1, 11, 111

0 01 110 111
a_1 a_2 a_3 a_4

9/02/19

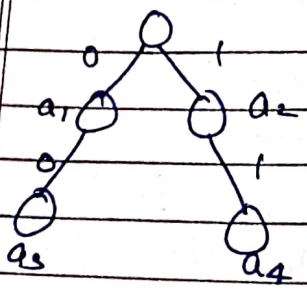
Prefix code
A code in which no codeword is prefix to another codeword is called prefix code

letter	code 1	code 2	code 3
a_1	0	0	0
a_2	1	10	01
a_3	00	110	011
a_4	11	111	0111

DATE / /201
PAGE

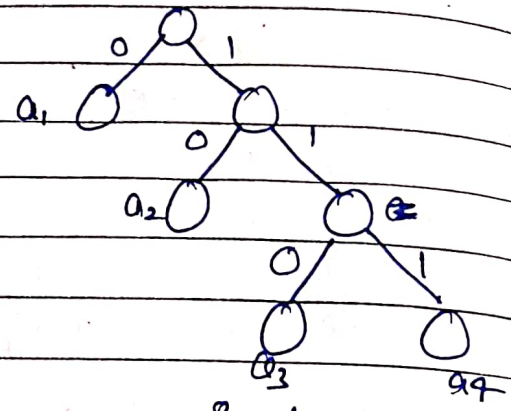
04/02

For code 1



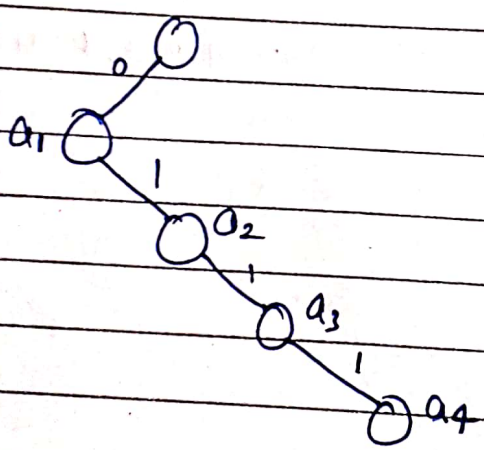
Not uniquely decodable

For code 2



uniquely decodable

For code 3



Not uniquely decodable

Internal node : Node that gives rise to other node is called internal node

External node : Node that doesn't give rise to other node.

04/02/21

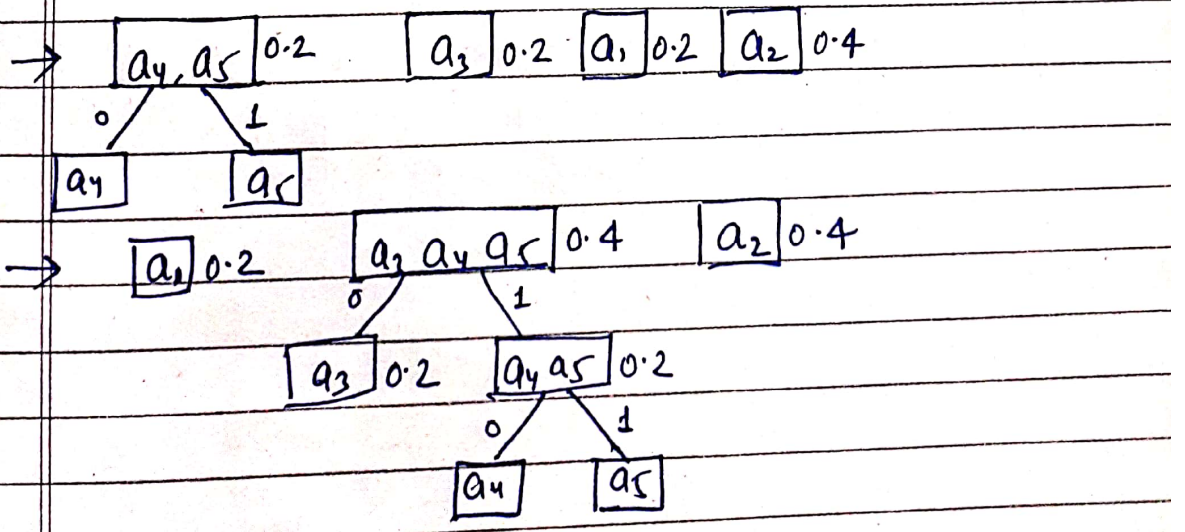
Huffman Coding

This technique developed by David Huffman is a part of class assignment.
The code generated using this technique are called Huffman codes. These codes are prefix code & optimum for given model
The Huffman procedure is based on two observation :-

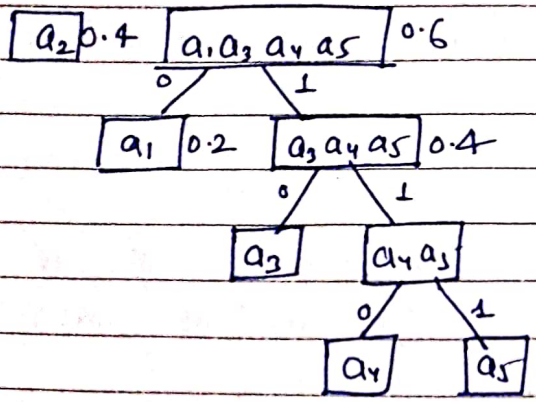
- In optimum code, symbols that occur more frequently will have shorter code words in comparison of that symbols occur less frequently.
- In an optimum code, the two symbols that occur least frequently will have the same length.

Q. Let $A = \{a_1, a_2, a_3, a_4, a_5\}$ with $P(a_1) = P(a_2) = 0.2$; $P(a_4) = P(a_5) = 0.1$; $P(a_2) = 0.4$
Find Huffman code & calculate entropy.

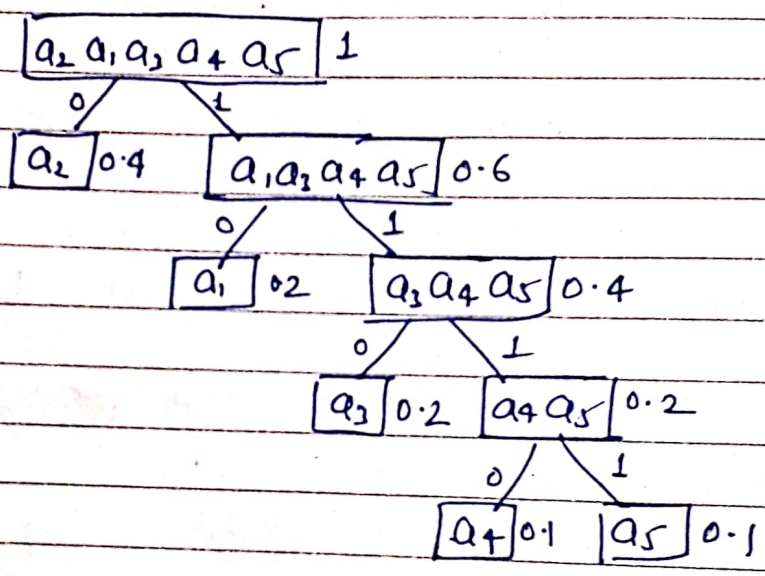
a_1	0.2	a_2	0.4
a_2	0.4	a_1	0.2
a_3	0.2	a_3	0.2
a_4	0.1	a_4	0.1
a_5	0.1	a_5	0.1



→



→



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

$= 0.4 + 0.4 + 0.6 + 0.4 + 0.4$

$= 2.2$ bits / ~~signal~~ symbol

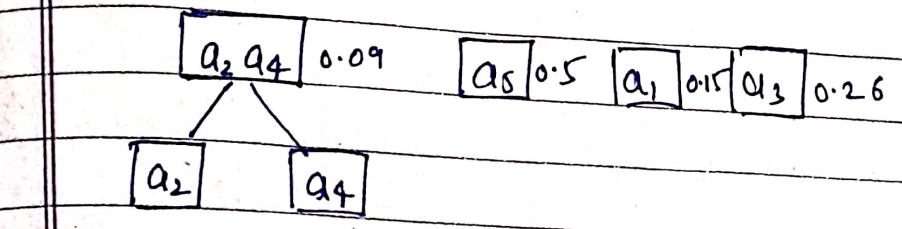
Entropy = $\sum_{i=1}^{n}$ Code length × P(E)

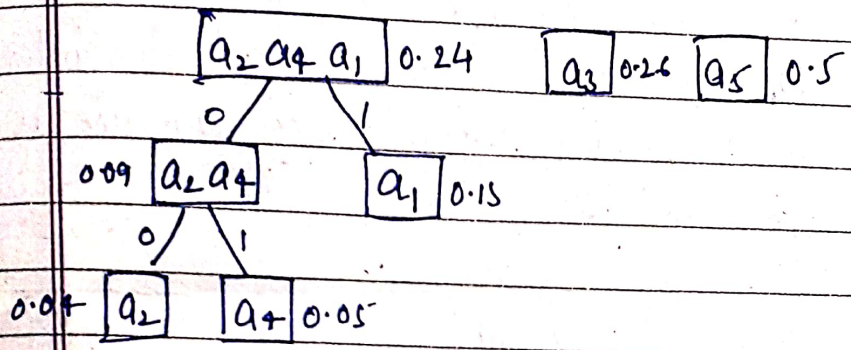
Q. A source ~~emits~~ emits letters from an alphabet $A = \{a_1, a_2, a_3, a_4, a_5\}$ with P(E) $p(a_1) = 0.15$, $p(a_2) = 0.4$, $p(a_3) = 0.26$, $p(a_4) = 0.05$, $p(a_5) = 0.5$

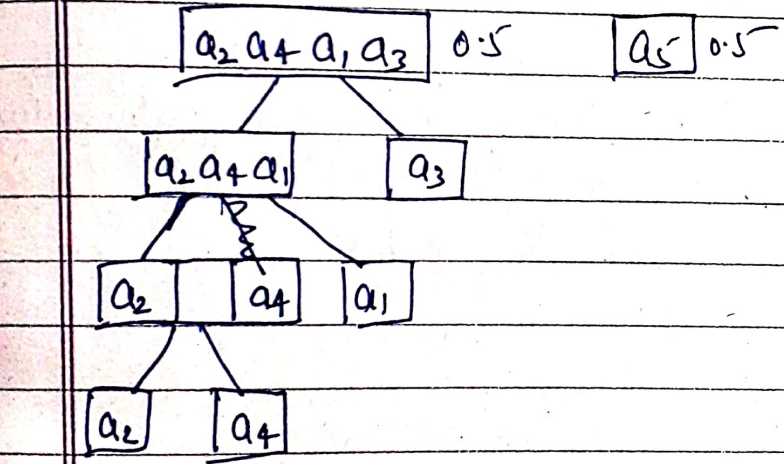
(i) Calculate H.C.

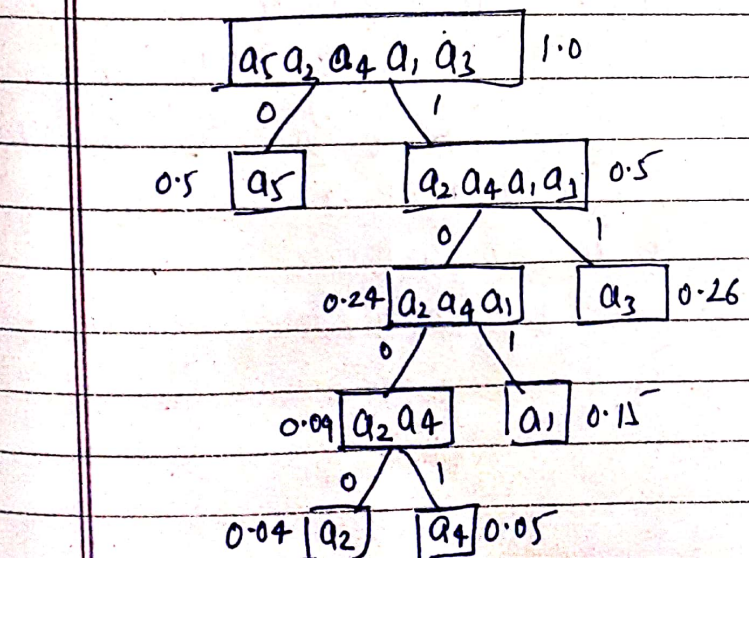
(ii) Find entropy

a_1	0.15	a_5	0.5
a_2	0.04	a_3	0.26
a_3	0.26	a_1	0.15
a_4	0.05	~~a_5~~	~~0.5~~
a_5	0.5	a_4	0.05
		a_2	0.04









DATE / /201
PAGE

Entropy = $0.5 \times 1 + 0.26 \times 2 + 0.15 \times 3 + 0.05 \times 4 + 0.04 \times 4$

$= 0.5 + 0.52 + 0.45 + 0.2 + 0.16.$

$= 1.02 + 0.45 + 0.36$

$= 1.83$

05/02/19

Adaptive Huffman Coding

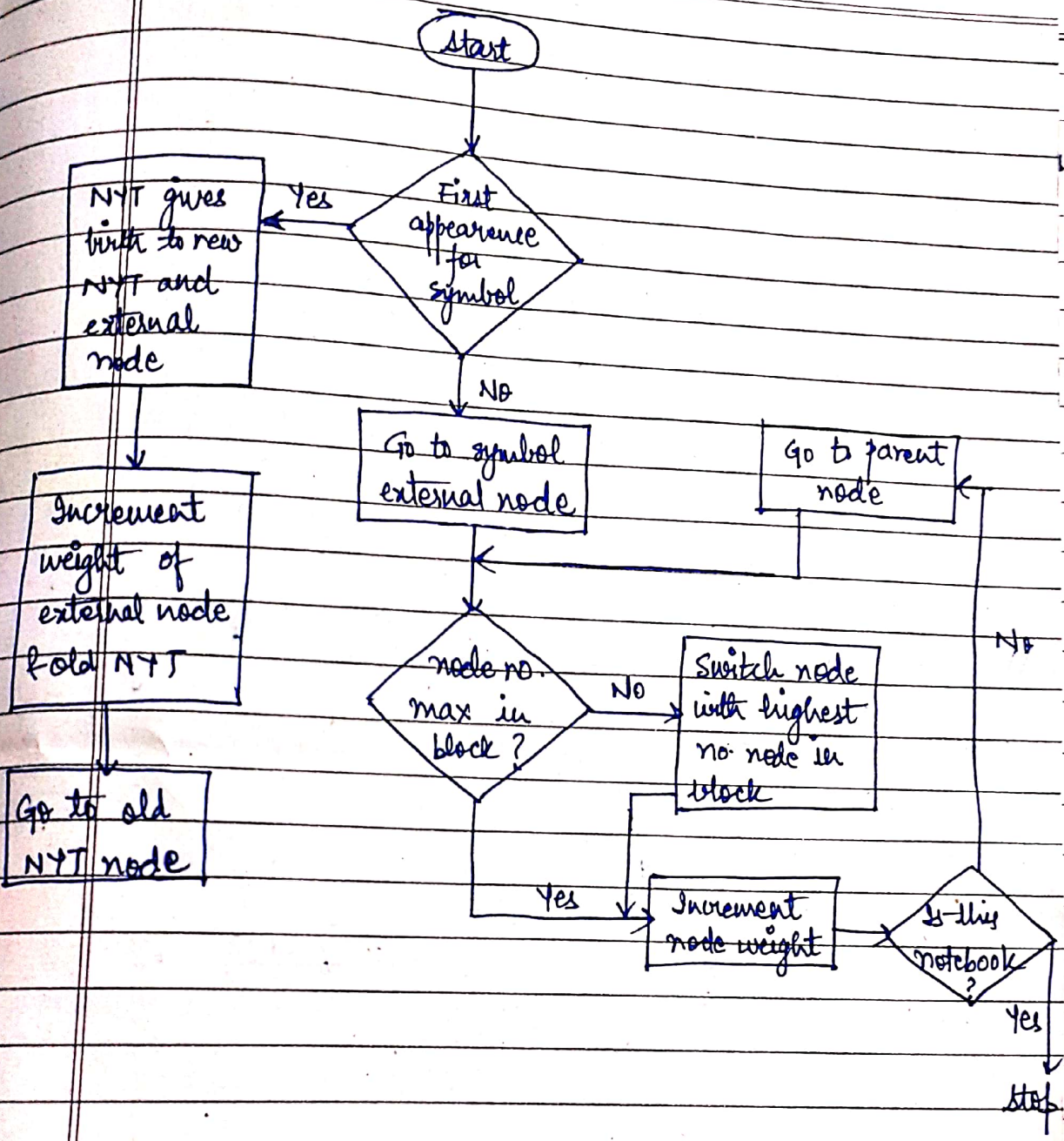
Huffman coding require knowledge of the P(E)'s of the source sequence if this knowledge isn't available we can use adaptive huffman coding.
In the adaptive huffman coding, neither transmitter nor receiver knows anything about source sequence at the start of transmission.

In this procedure, transmitter & receiver consist of a single node which is known as NYT(Not Yet Transmitted) node & weight of this node is 0.
As the transmission progress, nodes corresponding to symbols transmitted will be added to tree & tree is reconfigured using update procedure.

- ## Update Procedure

Update procedure requires that the nodes be in fixed order. This ordering is preserved by numbering the nodes. Largest code given to root node in the tree and smallest no. is assigned to NYT node.
No. from NYT node to the root are assigned in an increasing order from left to right & from lower to upper layer.

→



Tree Update (For encoding / for decoding)

Two Parameters : 1. weight of each leaf
2. Node no. (Every node have code no.)

weight of each leaf

For external node
⊡ weight
no. of time symbol encountered

For Internal node
⊙
It consist of sum of weight of its child

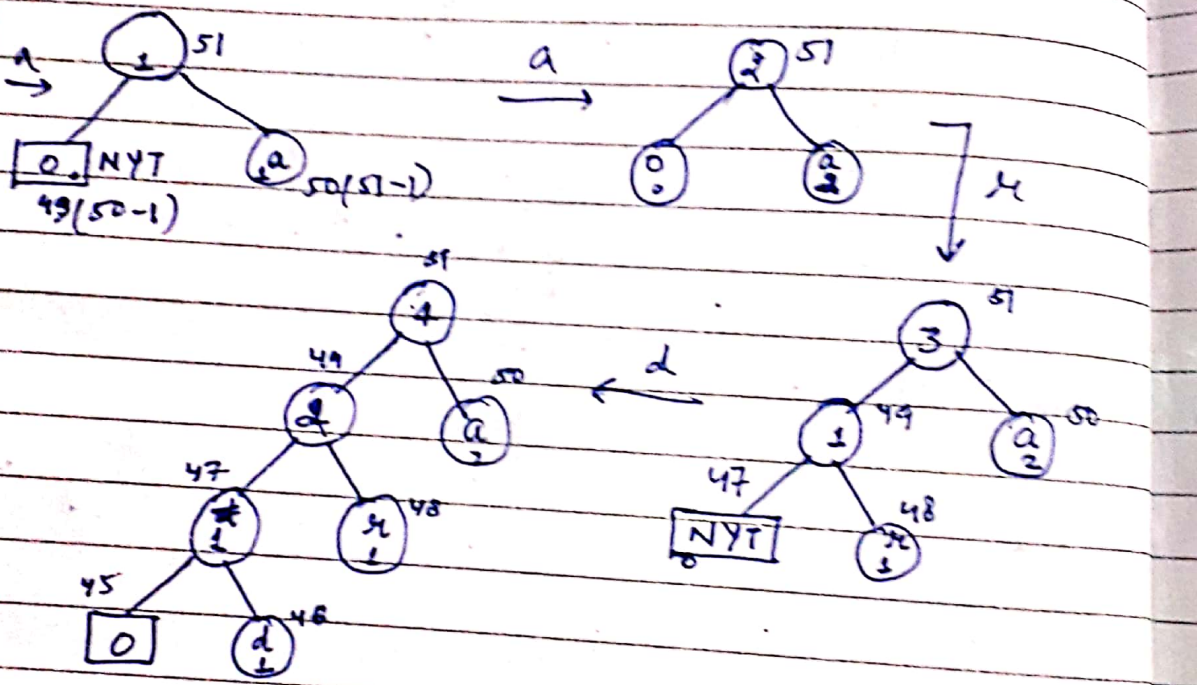
node no. = total no. of nodes in tree
= (2m-1)

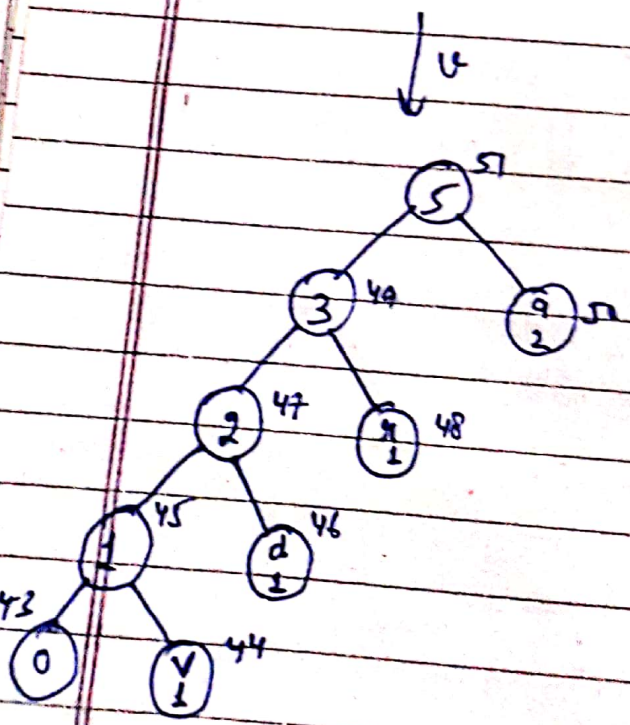
Encoding Procedure

"a e r d v a r k" encode with adaptive huffman coding.

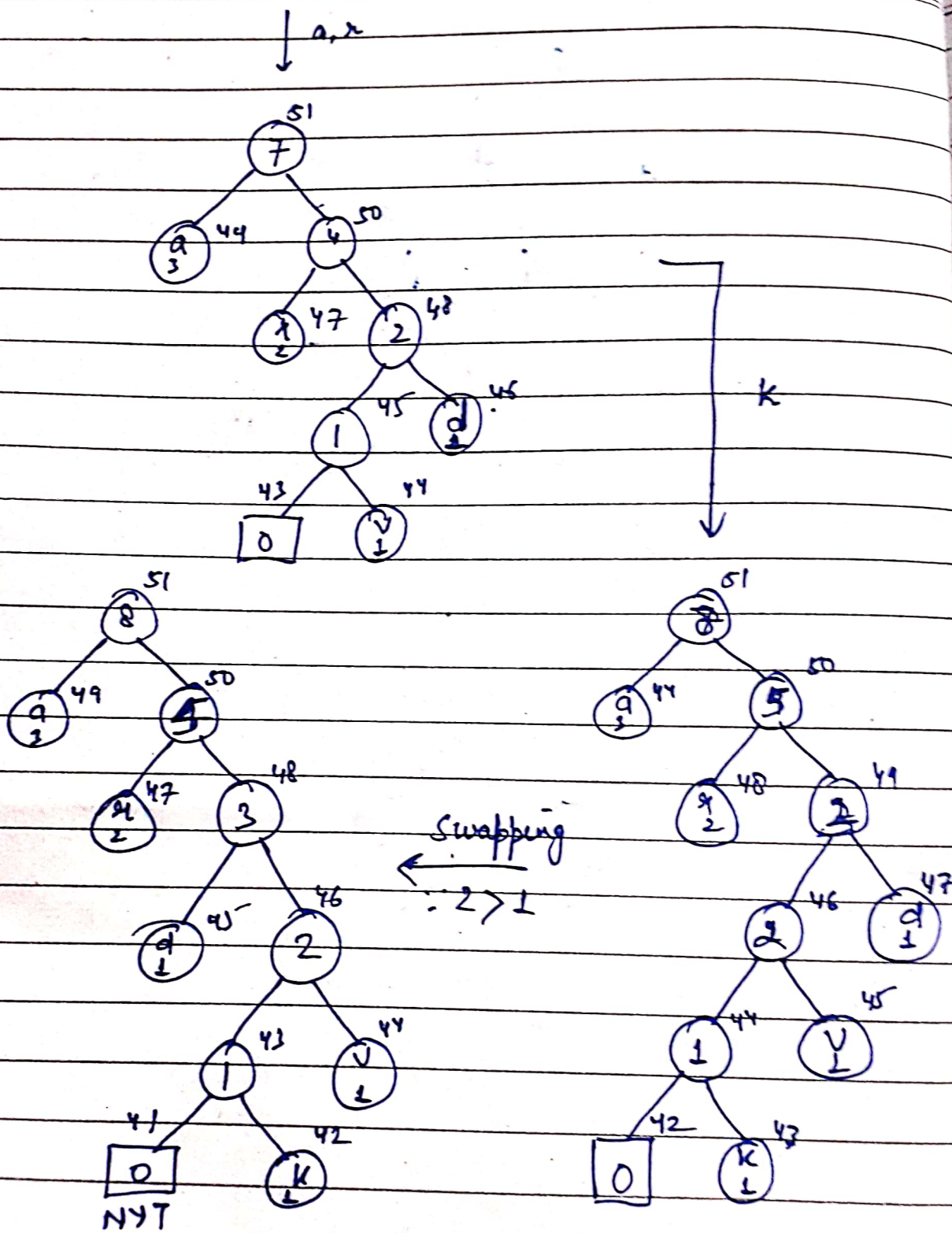
(i) 

(2m-1) where m = 26

(ii) 



Swapping when wt [left(child)] > wt [right(child)]



DATE / /201
PAGE

- **Adaptive Huffman Coding (Encoding Procedure)**

(i) If the symbol is encountered first time then we send the NYT code followed by fixed message as.

(ii) If symbol is already encountered, only send code for that symbol.

→ NYT code

Traversing the tree from root node to NYT node

→ Fixed code

for this we have two parameters e & r

$$m = 2^e + r.$$

m = no. of alphabet in dictionary.

$m = 26 = 2^e + r.$

where $e = 4, r = 10$ is fixed.

(i) If $1 \leq k \leq 2r$

(k: position of symbol in dictionary)

then symbol encoded as (e+1) bit binary representation of (k-1).

(ii) If $k > 2r$ then symbol encoded as 'e' bit binary representation of $(k - r - 1)$

Q. Encode "a a r d v a r k"

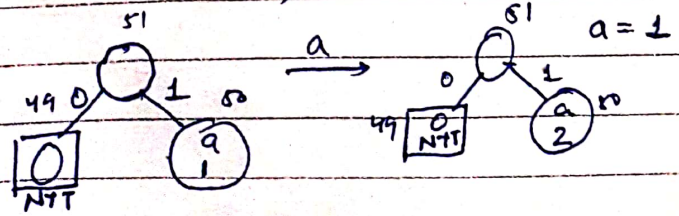
a - NYT - +

$1 \leq k \leq 2r$

$1 \leq 1 \leq 20$ (e+1) bit (k-1)

5 bit (1-1) ⇒ 5 bit 0.

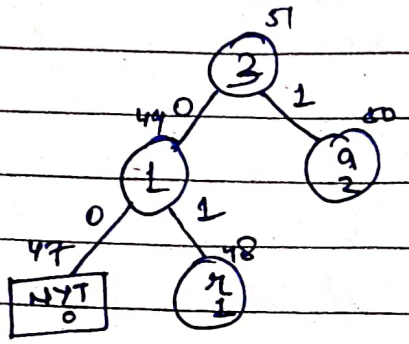
a = 00000



$r \rightarrow 1 \leq 18 \leq 20$

$r \rightarrow$ 5 bit $(18-1) \Rightarrow$ 5 bit 17

$r = 10001 \xrightarrow[NYT]{\text{adding}} 010001$

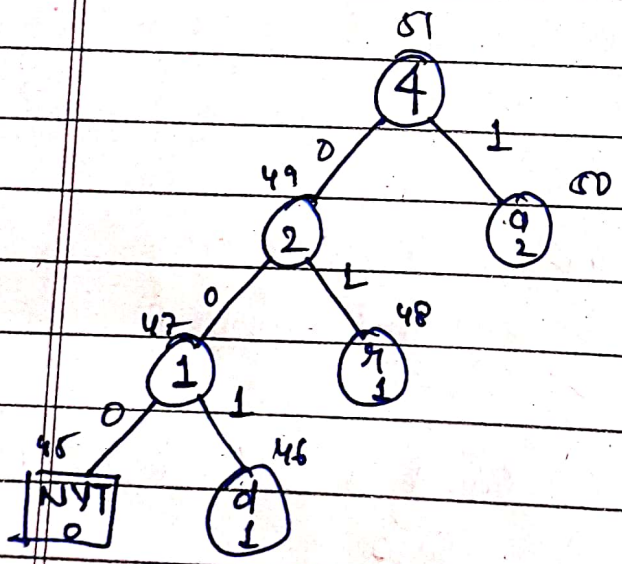


$d \rightarrow 1 \leq 4 \leq 20$

$d \rightarrow$ 5 bit $(4-1) \Rightarrow$ 5 bit $\neq 3$.

$d = 00011$

$d = 0\ 00011$

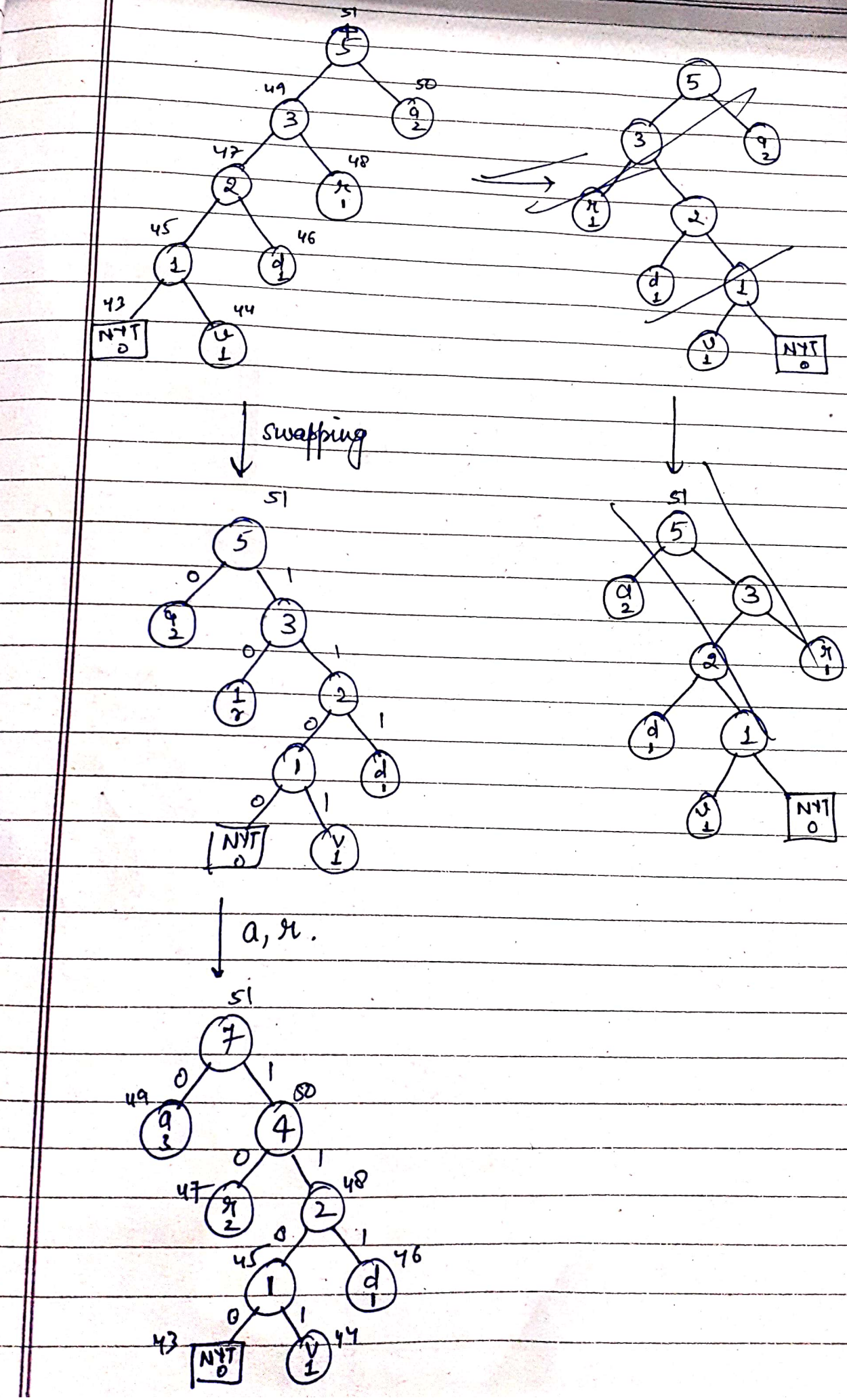


$v \rightarrow 22 > 20$ (Case ii)

$v \rightarrow 22 - 10 - 1 = 11$ in 4

$v \rightarrow 1011$

NYT $\rightarrow$ 000.



$K \rightarrow 21 \leq 11 \leq 20$

k → 5 bit 10

K → 01010

NYT → 1100.