DCDR

Meghpara Jeet J
160410116056
T4IT-1.

* ASSIGNMENT 3 *

॥ नमामि देवी सरस्वति ॥

1. Explain Huffman code in detail. Define minimum variance.

→ The code generated in which codes are prefix codes & are optimum for given model.

Based on two observation:

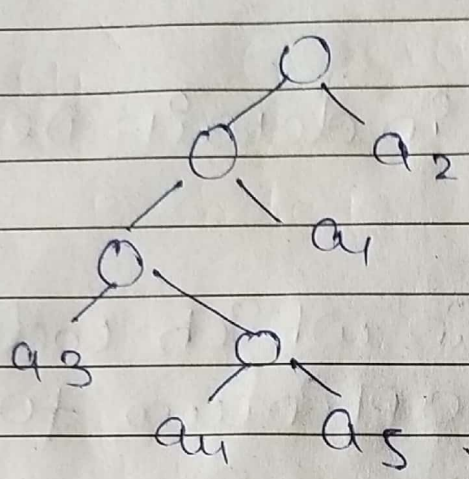
(a) In optimum code, sym that occur more frequently will have shorter codewords than sym that occur less frequently.

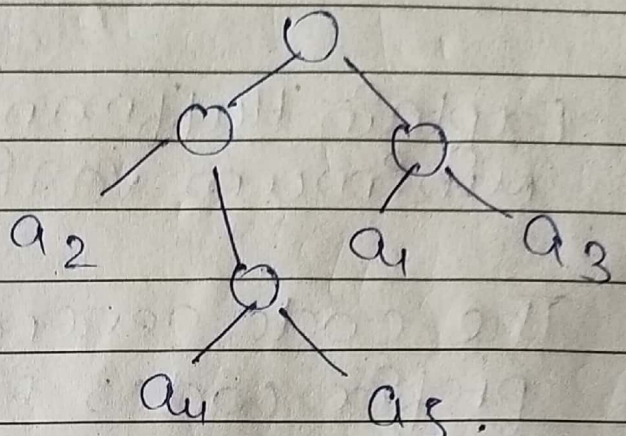
(b) two sym that occur least frequently will have same length.

Minimum variance huffman code

→ In this method technique first combined word is placed at highest priority rather than lowest

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





2. List out rules of Huffmann coding with example of -

	Counts
A	14
B	7
C	5
D	5
E	4

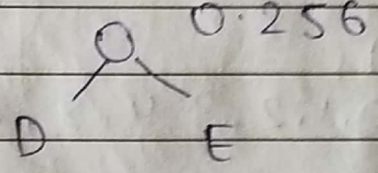
→ Rules.

(a) In optimum code, sym that occur more frequently will have shorter codeword.

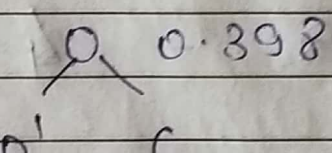
(b) two sym that occur least frequently will have same length codeword.

Total counts = 35.

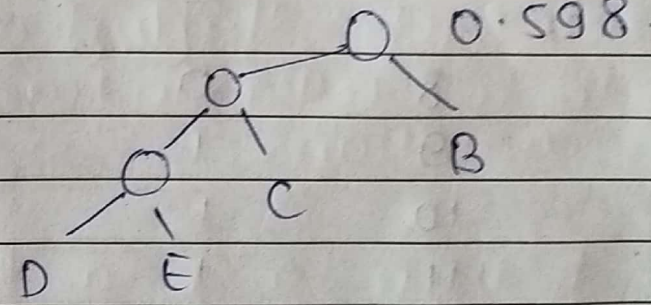
A	0.4
B	0.2
C	0.142
D	0.142
E	0.114



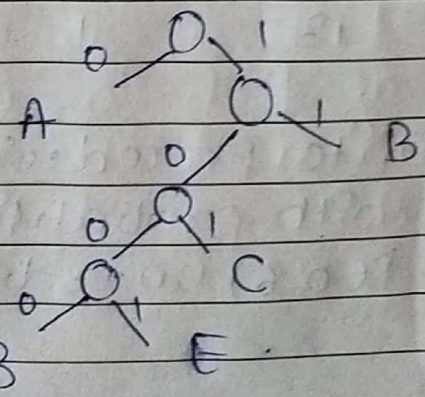
A	0.4
B	0.2
D'	0.256
C	0.142



A	0.4
C'	0.3982
B	0.2



A	0.4
B'	0.598



A	- 0
B	- 11
C	- 101
D	- 1000
E	- 1001

Q4. Generate GOLOMB code for m = 9 and n = 8 to 13.

→ $\lceil \log 9 \rceil = 4$

$\lfloor \log 9 \rfloor = 03$

For diff code of 9.

$= 2^{\lfloor \log_2 m \rfloor} - m$

$= 2^4 - 9$

$= 7 \ (0,1,2,3,4,5,6)$ - 3 bits

rest values

9 + 2 - 4 bits

m=9

n	q	r	unary(q) + binary(r) codeword
8	0	8	01000
9	1	0	10000
10	1	1	10001
11	1	2	10010
12	1	3	10011
13	1	4	10100.

Q5. Write procedure to generate TUNSTALL code with probability P(A) = 0.3 P(B) = 0.1 P(0.6) n = 3 bits.

→

A	0.6
B	0.3
C	0.1

Here, A has highest probability we remove it & add two string

B	0.3
C	0.1
AA	0.36
AB	0.18
AC	0.06

Still it is less than 2^3 codewords we go on, we get another 2

B	000
C	001
AB	010
AC	011
AAA	100
AAB	101
AAC	110

→ and we stop here as $7 < 2^3$ and catch on it will increase in next iteration

Q.6. Explain Burrows-Wheeler Transform algo.

→ It is used for loss less compression.

→ It transform msg in such a way which is more convient for compression.

→ It form cluster of common occuring letters so that you can compare it easily.

Steps. 1. Make list of all rotation / probabilities of given string
2. Sort them
3. Take last column.

→ final column is our output.

Eg. Create cyclic shift of original sequence, as there 7 letters in banana$ there will be 7 permutations.

Step 1.	b	a	n	a	n	a	$	1
	$	b	a	n	a	n	a	2
	a	$	b	a	n	a	n	3
	n	a	$	b	a	n	a	4
	a	n	a	$	b	a	n	5
	n	a	n	a	$	b	a	6
	a	n	a	n	a	$	b	7

Step 2:

Sort them

$ b a n a n a	1
a $ b a n a n	2
a n a $ b a n	3
a n a n a $ b	4
b a n a n a $	5
n a $ b a n a	6
n a n a $ b a	7

Output

last column

annb$aa