Class Partipation – No credit.

Hufmann Encoding

– Why we study it.
- Formal defⁿ
- Greedy Algorithm

Encoding: Representing symbols (in message) using binary codes 0/1.

Eg. msg = CABDC

SYMBOL	CODES
A	00
B	01
C	10
D	11

2 bit string

Eg. In computer code (ASCII co

A – 01 00001 (65)

B – 0 1 00010 (66)

⋮

8 bit string

"Variable length Coding" – All codes need not be of same length.

Eg.

Table →

SYMBOL	CODES
A	0
B	100
C	101
D	11

length ∈ [1, 3]

Question – When can such encodig be useful.

Eg. freq: A – 45, B – 9, C – 11, D – 35 in a msg of 100 char.

100 = 45 + 9 + 11 + 35

Length of encoded msg. = 45(1) + 9(3) + 11(3) + 35(2)

= 175 ← 12% impro

If we were usig $\underbrace{00, 01, 10, 11}_{len=2}$ as encodig, then length = 200

Is there ambiguity?

msg = CABDC → $\overset{C}{\overset{\Uparrow}{101}}$ 0 100 0 11 ← enco msg. = CABAD → 101 0 100 0 11 ← enc

Not mapped

Not mapped

mapped to C

To take care of ambiguity we ensure

if $(x_1 \dots x_k)$ is CODE-WORD, then no prefix of it is co

Prefix Encoding : def^n

Eg. Country call nos.

+91_

X

+91 – India

+1 – USA. (10 digit)

+19 – Not a country code. (9 digit)

Ambiguity

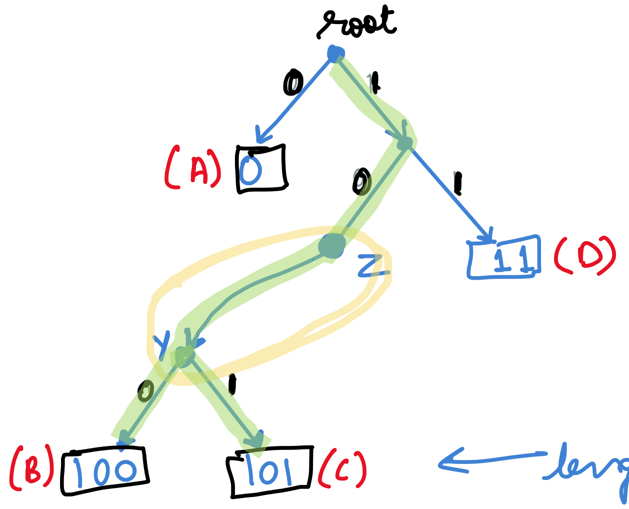
Huffman Encod-

⊛ JPEG/MP3 compressi

⊛ Zipping a file

Tree Representation

	CODES
A	0
B	100
C	101
D	11



← length = ~~4~~ → 3

Property of Prefix

→ All symbols mu be leaf nodes

Problem:

Given : Symbols $(a_1 \dots a_n)$ with freq. vector $F = (f_1 \dots$

Find : Prefix encoding for which encoded msg has "min"

A binary tree T with leaves $(a_1 \dots a_n)$ such that

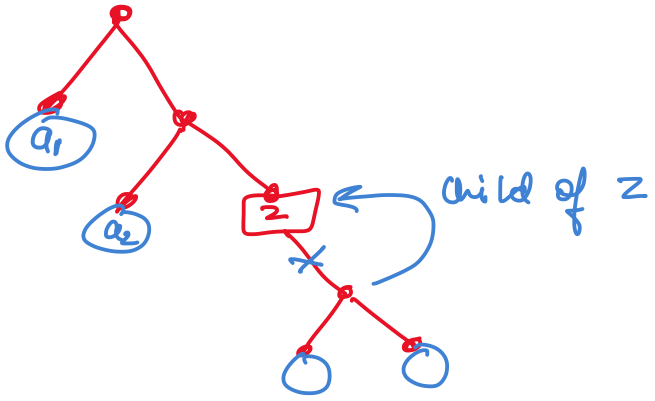
$\left(\sum_{i=1} (f_i)(\text{depth of } a_i \text{ in } T)\right)$ be m

Property 1 : Each internal node should have 2 children

Proof : By contradiction

Take a tree T which is not complete & let

internal node of degree 1.



→ All leaf still co

$\sum_{i=1} f_i$ * is re

Ques : Is it necessary that one child of each no

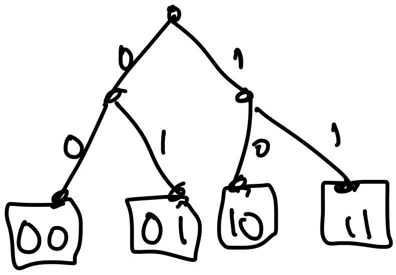
No.

A – 25

B – 25

C – 25

D – 25



Property 2 : If $f_1 \geq f_2 \geq \dots \geq f_n$. Then in o

(i) $\text{depth}(a_1) \leq \text{depth}(a_2) \leq \dots \leq \text{depth}(a_n)$

(ii) $\text{depth}(a_n) = \text{depth}(a_{n-1})$

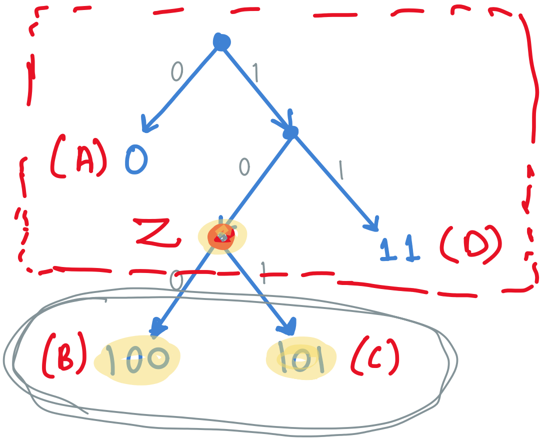
Proof of (ii) : Both children of parent (a_n)

MA~~X~~ —

are leaf node, & will | deeper
correspond to two minimum frequencies |

Z

	CODES
A	0
~~B~~	100
~~C~~	101
D	11



F

A	45
B	9
C	11
D	35

Greedy strategy →

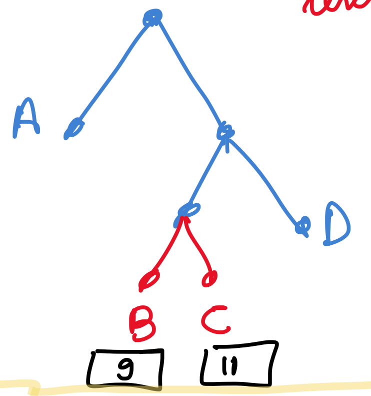
Take minimum 2 freq & replace them with new freq $f = f_{min} + f_{secondmin}$

F′

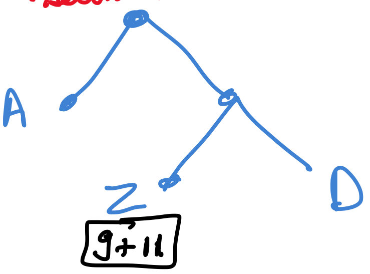
A	45
Z	20
D	35

↓ tree T′

T



←



$$|\text{encoded-msg}(F)| = |\text{encoded-msg}(F')| + \underbrace{(1)}_{\text{diff in le}} ($$

H.W. – Correctness of this algo.

Theorem: If a_{n-1}, a_n have least frequency, then
problem

$F^* = (F \cup \{f$

$F = (F \cup \{f_n + f_{n-1}\}) \setminus \{f_n, f_n$

$$opt(F) = opt(F^*) + (f_{n-1} + f_n).$$

<u>H.W.</u> – Find an $O(n \log n)$ time implementation.