

# CS & IT ENGINEERING

## Algorithms

Greedy Method

Lecture No. - 03



By- Dr. Khaleel Khan

sir

# Recap of Previous Lecture



Topic

KNAP Problem

JSD

Optimal Merge Pattern

# Topics to be Covered



Topic

Optimal Merge Patterns

Huffman Coding

Spanning Trees



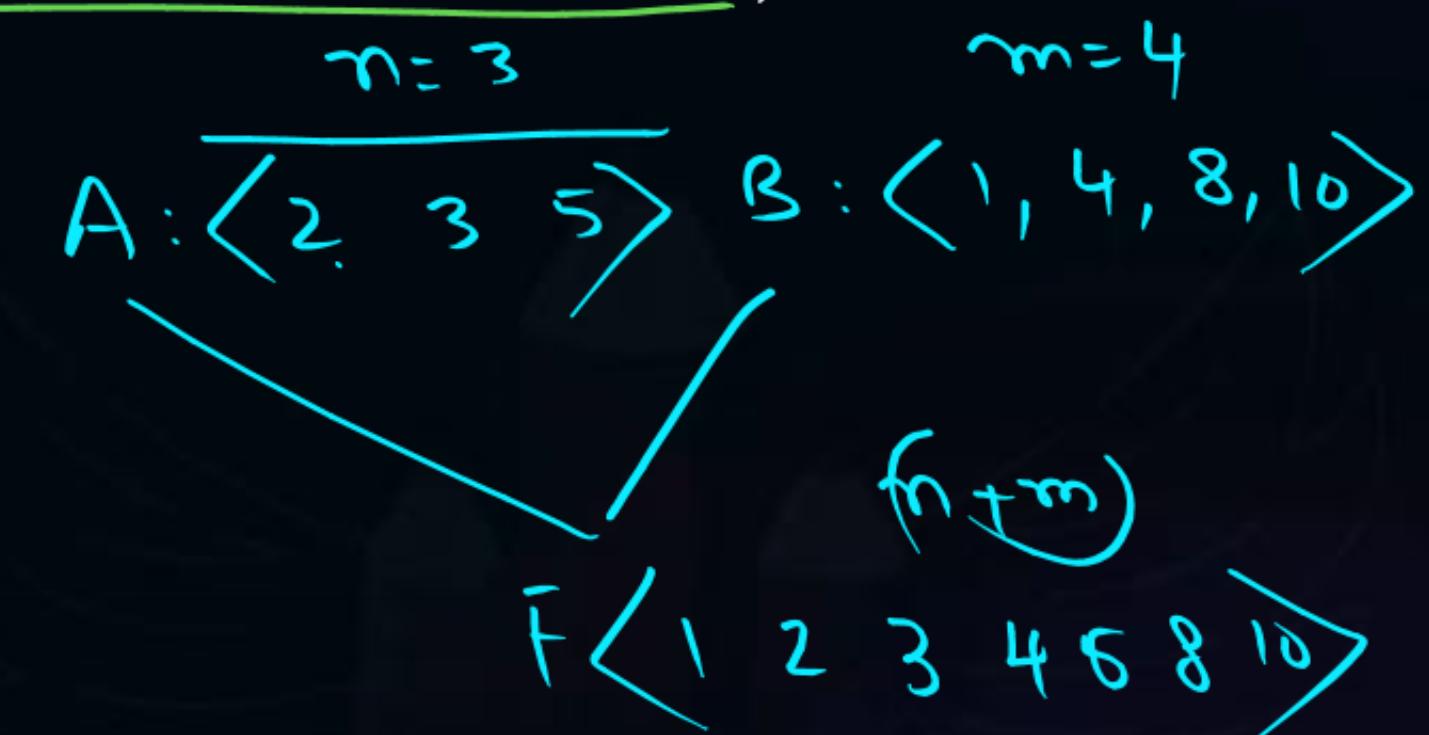
## Topic : Greedy Method



### 3) Optimal Merge Patterns:

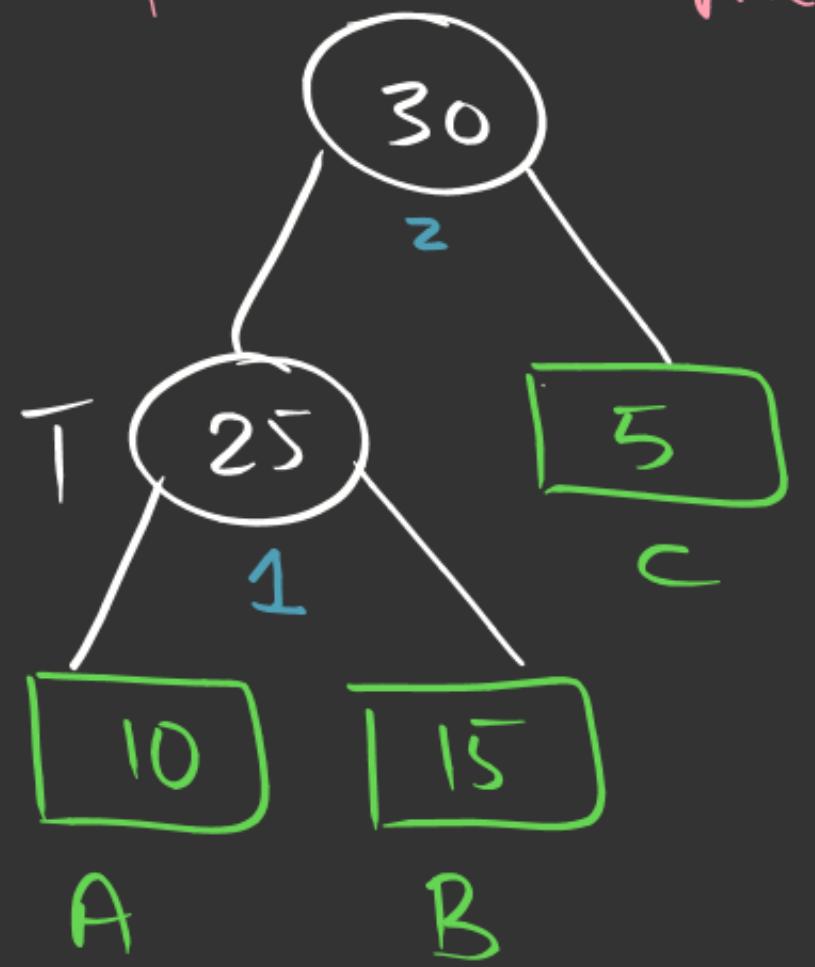
→ Given Two files 'A' & 'B' having 'n' & 'm' records respectively; To Merge them , using 2-way Merging it requires  $(n+m)$  record movements;

→ Given 'n'-files, it is req'd to Merge them using 2-way Merging , such that, the total no. of record Movements is Minimum,

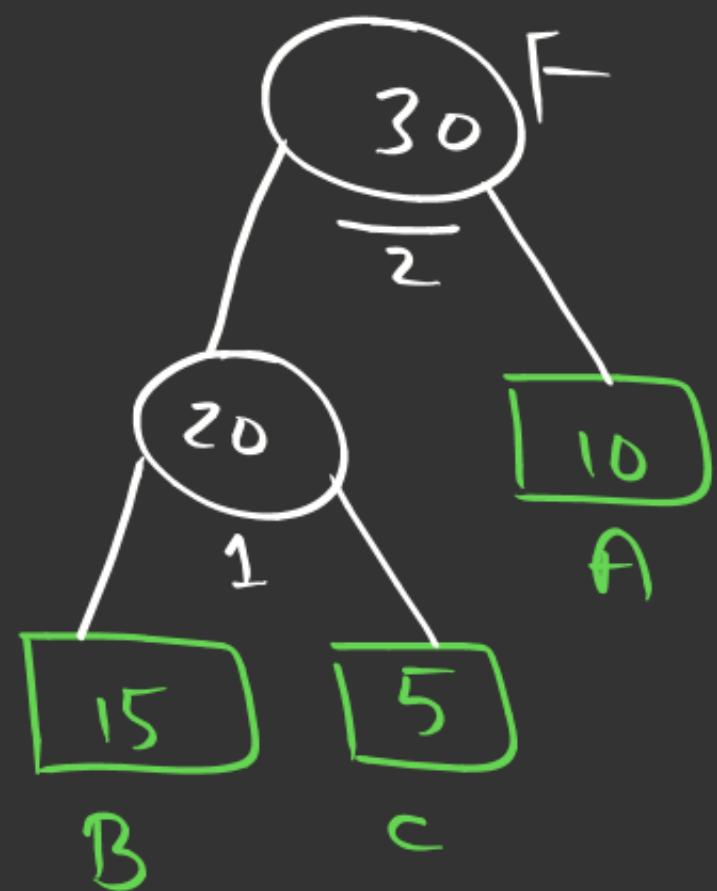


$$A: \underline{10}; \quad B: \underline{15}; \quad C: \underline{5}$$

(2-Way Binary Merge Tree)



$$: 25 + 30 : \underline{\underline{55}} \\ \langle A \ B \ C \rangle$$



$$: \underline{\underline{20+30}} : 50 \\ \langle B \ C \ A \rangle$$

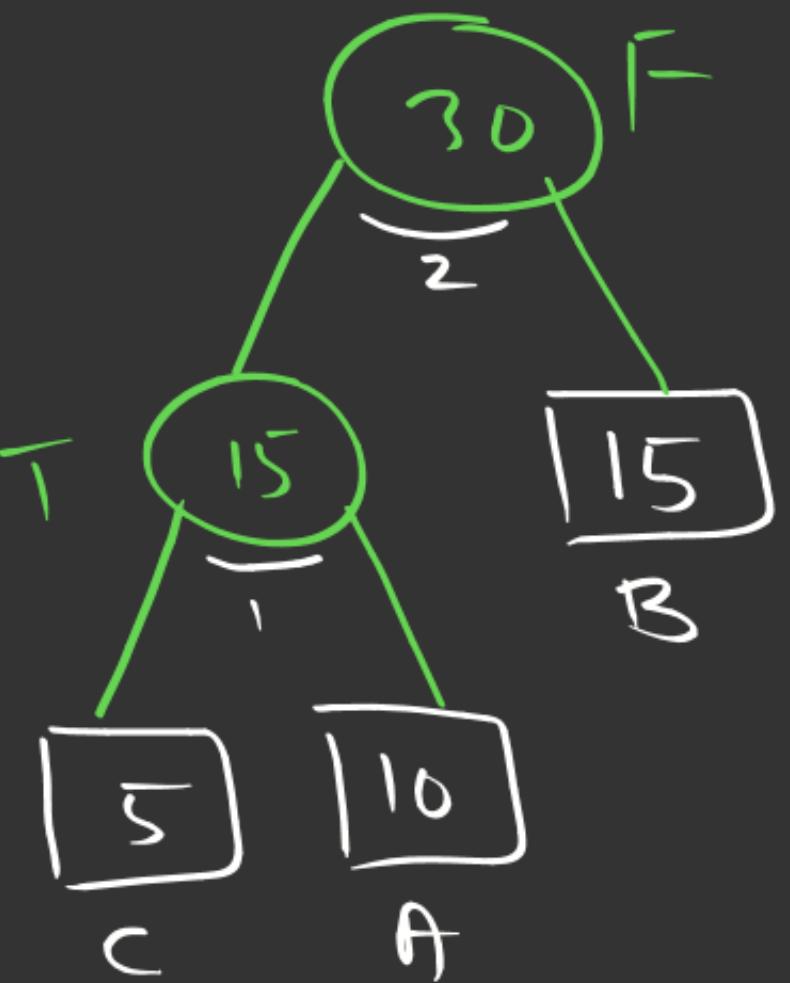
For  $n$ -files, it requires  $(n-1)$  Merge ops

→ We want the pattern which is generating min no. of record movements

→ Soln Space:  $\underline{\underline{n!}}$

## Greedy Strategy:

$$\frac{A: 10}{F_1}, \quad \frac{B: 15}{F_2}, \quad \frac{C: 5}{F_3}$$



Total record :  $15 + 30 = 45$   
Mov's

The No. of record Mov's  
= Weighted External Path length

$$\text{Wtd. Ext. Path length} = \sum_{i=1}^n d_i * q_i;$$

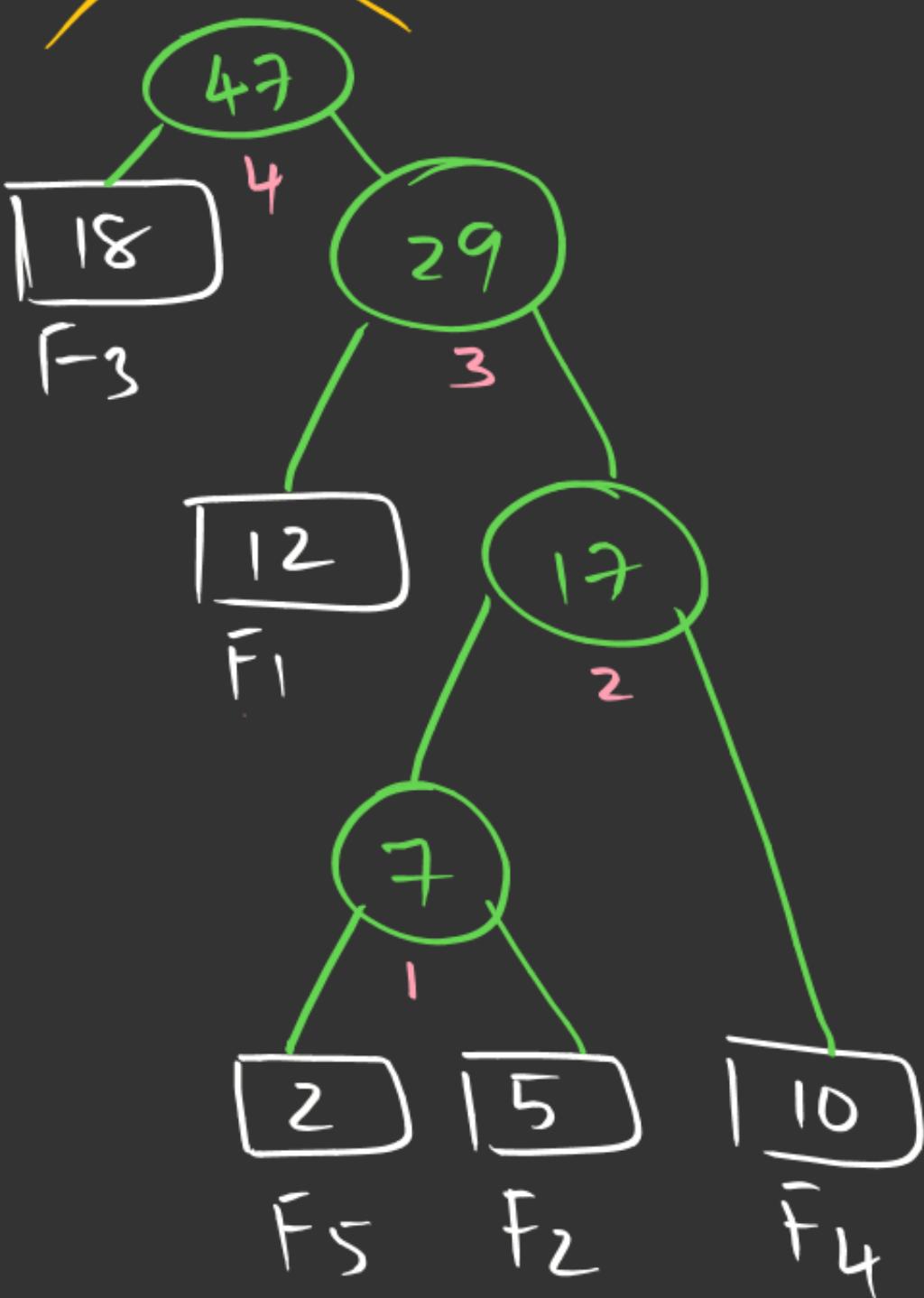
$d_i$  = distance from root to  $F_i$

$q_i$  = freq / size of  $F_i$ ;

$$= 2 * 10 + 1 * 15$$

$$= 20 + 15 + 10 = \frac{+ 2 * 5}{45} = 45 //$$

$$\langle F_1, \dots, F_5 \rangle = \langle 12; 5; 18; 10; 2 \rangle$$



No. of record Mov's:

$$7 + 17 + 29 + 47 = \underline{\underline{100}}$$

wtd-Ext-Path

$$\text{Length} = \sum_{i=1}^n d_i * v_i$$

$$\begin{aligned}
 &= 2 * 12 + 4 * 5 + 1 * 18 \\
 &\quad + 3 * 10 + 4 * 2
 \end{aligned}$$

$$= \underline{\underline{100}}$$

Optimal 2-way Binary  
Merge Tree



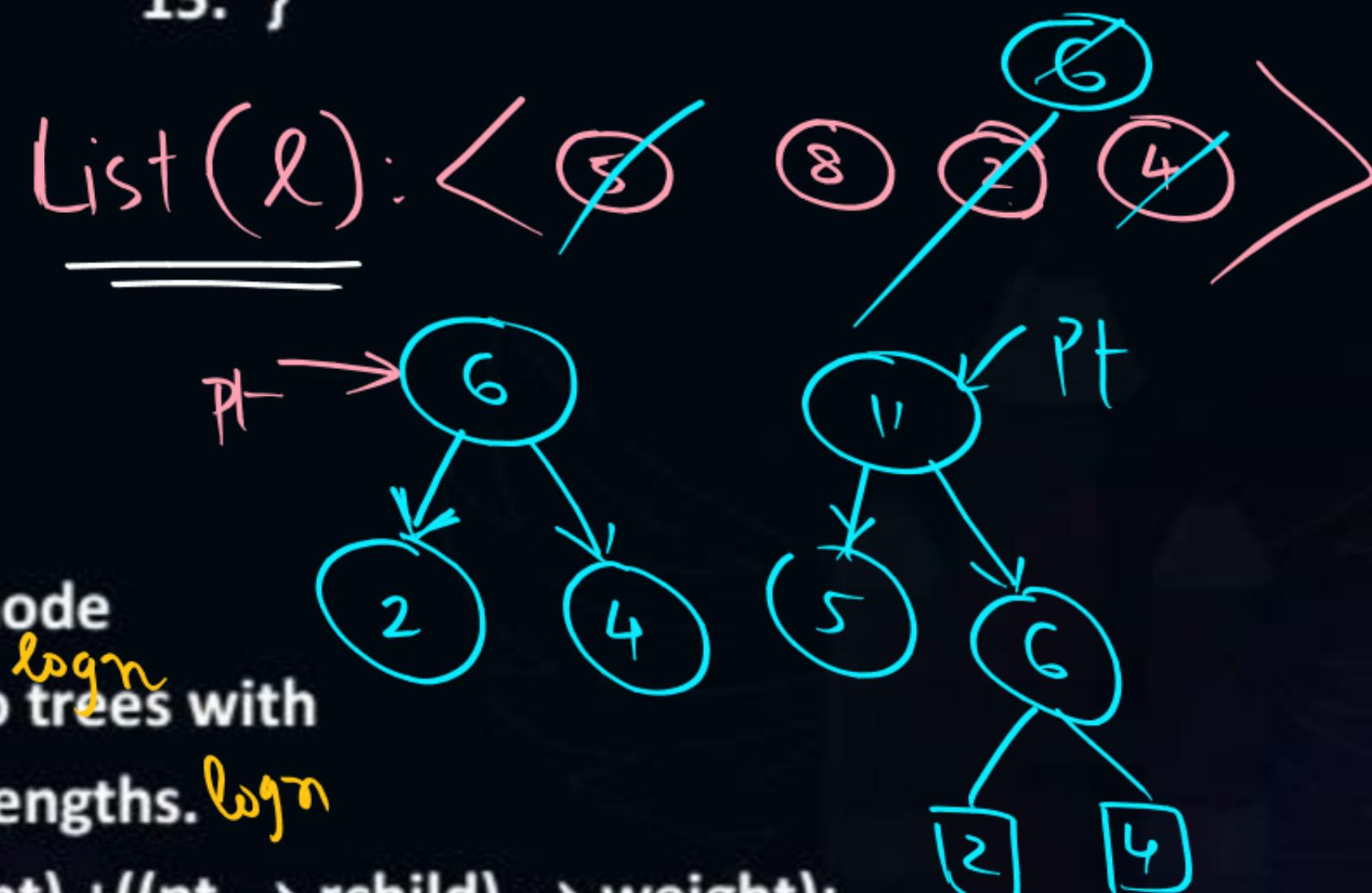
# Topic : Greedy Method



```
treenode = record {  
    treenode * lchild; treenode * rchild;  
    integer weight;  
};
```

1. Algorithm Tree(n)
2. // list is a global list of n single node
3. // binary trees as described above.
4. {
5. for i := 1 to n - 1 do :( $n-1$ )
6. {
7. 1) pt := new(treenode); // Get a new tree node
8. 2) (pt → lchild) := Least(list); // Merge two trees with  $\log n$
9. 3) (pt → rchild) := Least(list); // smallest lengths.  $\log n$
10. 4) (pt → weight) := ((pt → lchild) → weight) + ((pt → rchild) → weight);

12. Insert(list,pt);
13. }
14. return Least(list); // Tree left in list is the merge tree.
15. }



1) Time Complexity : for  $n$ -file:

→ Impl. of list

2) Space Complexity

Input : ' $n$ '

Space Complexity

:  $O(n)$

Nodes

Linear  
linked list  
' $n$ '

Least : 'n'  
Insert: 1       $O(n^2)$

Heap (Non-linear) : 'n' elements

— Insert :

— delete :

— dec|Inc :  
key

$O(\log n)$

Time Complexity =  $O(n \cdot \log n)$

# Huffman Coding: <An Application of Opt. Merge Pattern>

(Data Encoding + Data Compression Technique)

↓  
repr. an  
element with  
some code

→ 4 characters/Elements

→ Straight forward: 2 bits  
Normal Encoding (00 - a  
01 - b  
10 - c  
11 - d)

→ 100 characters  
→ 7 bits/character  $\lceil \log_2 N \rceil$  bits

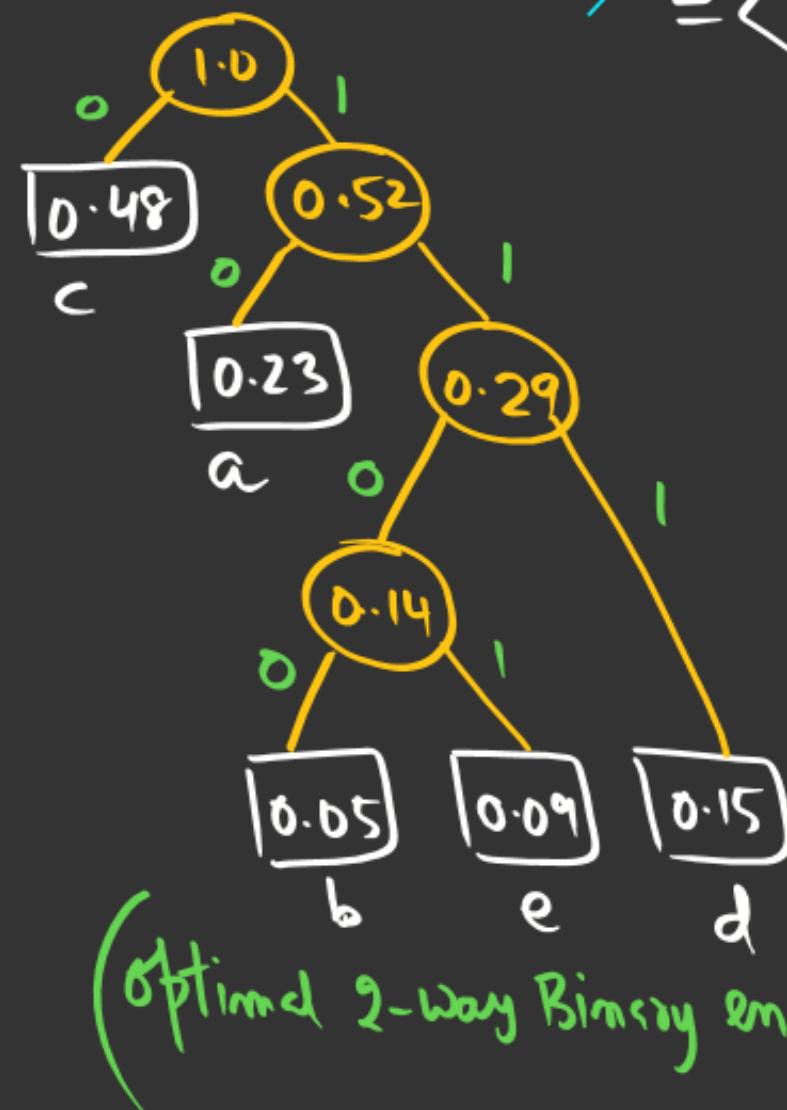
→ Data Compression is  
Never possible with uniform  
encoding

→ Compression is possible with non  
uniform encoding

→ Coders are assigned based on frequency; [Non-Uniform encoding]

### Huffman Coding

$$L = \sum (a, b, c, d, e) = \langle 0.23; 0.05; 0.48; 0.15; 0.09 \rangle$$



: 10010001010001111...

: acaccaaccd...

- a → 10 (2 bits)
- b → 1100 (4 bits)
- c → 0 (1 bit)
- d → 111 (3 bits)
- e → 1101 (4 bits)

→ Uniform Encoding: 3 bits/char.

$$\frac{\text{Avg. No. of bits}}{\text{H.C.}} = \sum_{i=1}^n di * q_i$$

$$= 2 * 0.23 + 4 * 0.05 + \\ 1 * 0.48 + 3 * 0.15$$

$$+ 4 * 0.09 = \underline{\underline{1.95 \text{ bits}}}$$

Opt Morse  
Rttern  
Algo

Text: <ccacccabdc> IP

(i) Uniform encoding:  $10 * 3 = 30 \text{ bits}$

(ii) Huff Coding:

< 10100001011001110 >: 17 bits

ccacccaa



## Topic : III. Greedy Method



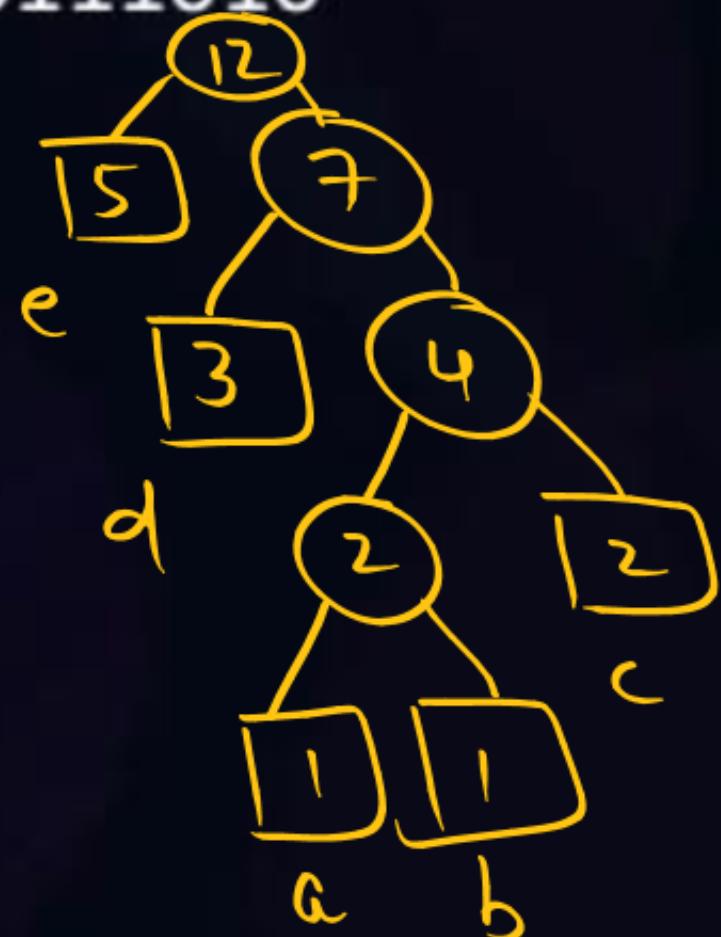
Q. The characters 'a' to 'h' have the set of frequencies based on the first 8 Fibonacci numbers as follows :

a:1, b:1, c:2, d:3, e:5, f:8, g:13, h:21

A Huffman code is used to represent the characters. What is the sequence of characters corresponding to the following code?

110111100111010

- (a) fdheg
- (c) dchfg



- (b) ecgdf
- (d) fehdg

a →  
b →  
c →  
d →  
e →  
f →  
g →  
h →

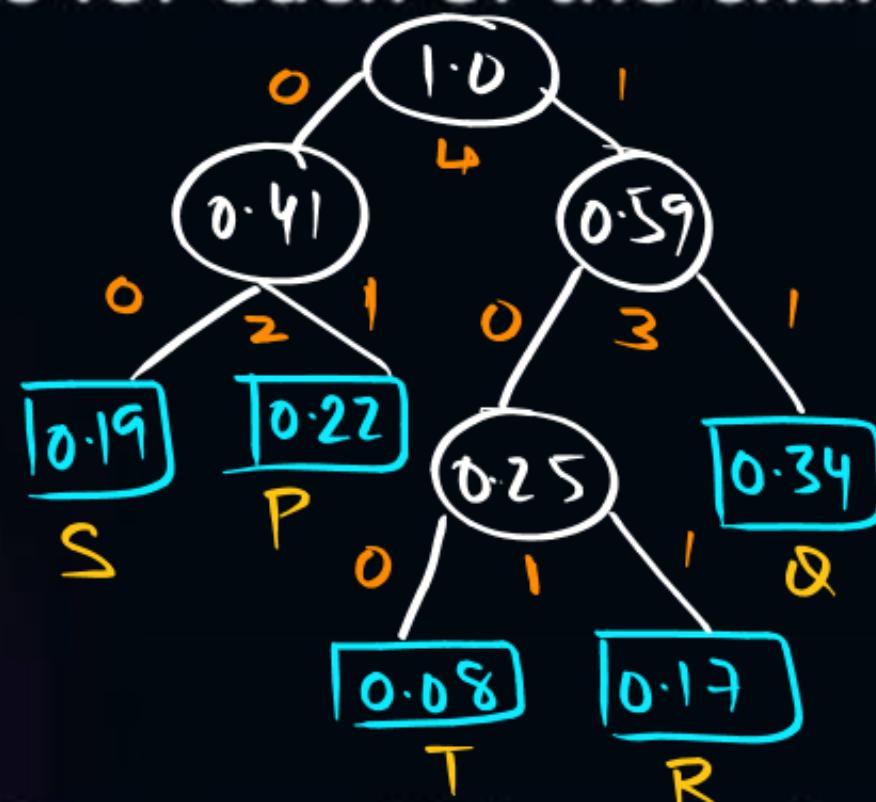


## Topic : III. Greedy Method



- Q. A message is made up entirely of characters from the set  $X = \{P, Q, R, S, T\}$ . The table of probabilities for each of the characters is shown below:

Character	Probability
P	0.22
Q	0.34
R	0.17
S	0.19
T	0.08
Total	1.00



$\left\{ \begin{array}{l} P \rightarrow 01 \quad (2) \quad \{0, 1, 01\} \\ Q \rightarrow 11 \quad (2) \quad \{1, 1, 11\} \\ R \rightarrow 101 \quad (3) \\ S \rightarrow 00 \quad (2) \\ T \rightarrow 100 \quad (3) \end{array} \right. \quad \{1, 10, 100\}$

If a message of 100 characters over  $X$  is encoded using Huffman coding, then the expected length of the encoded message in bits is (22.5) ✓

$$\sum_{i=1}^n d_i * a_{V_i} = (2 * 0.22 + 2 * 0.34 + 3 * 0.17 + 2 * 0.19 + 3 * 0.08) = 2.25$$



## Topic : III. Greedy Method

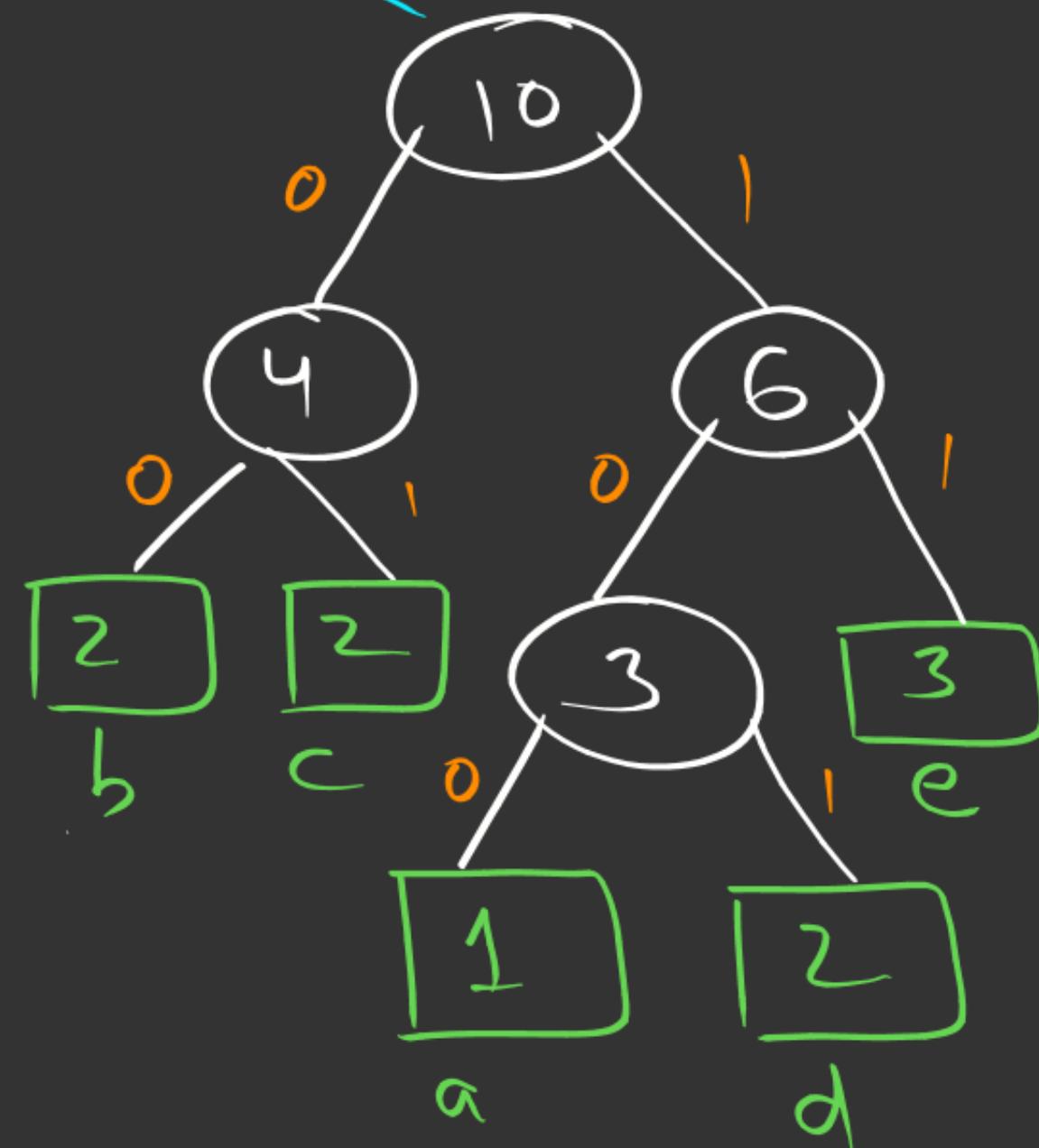
Q. Consider the string abbccddee. Each letter in the string must be assigned a binary code satisfying the following properties:

- H.C.
1. For any two letters, the code assigned to one letter must not be a prefix of the code assigned to the other letter.
  2. For any two letters of the same frequency, the letter which occurs earlier in the dictionary order is assigned a code whose length is at most the length of the code assigned to the other letter.

Among the set of all binary code assignments which satisfy the above two properties, what is the minimum length of the encoded string?

- (a) 25
- (b) 23
- (c) 21
- (d) 30

a b b c c d d e e e



$$\left\{ \begin{array}{l} a = 1 \\ b = 2 \\ c = 2 \\ d = 2 \\ e = 3 \end{array} \right.$$

a → 100	(3)
b → 00	(2)
c → 01	(2)
d → 101	(3)
e → 11	(2)

$$b \rightarrow 00 \quad (2)$$

C → O | (2)

$$d \rightarrow 101 \quad (3)$$

$$e \rightarrow l l \quad (2)$$

## 5) Minimum Cost Spanning Tree (MCST)

(Graph based Problem)

Graph (Nm-linear D.S.)

$$\hookrightarrow G = (V, E)$$

Undirected  
Directed

weighted  
Unweighted

$|V| = n$  (vertices)

$|E| = e$  (edges)

## Representation of Graphs:

- 1) Adjacency Matrix (Sequential)  
Array
- 2) Adjacency list (list based)

# THANK - YOU