CS & IT ENGINEERING

Algorithms

Miscellaneous Topics

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

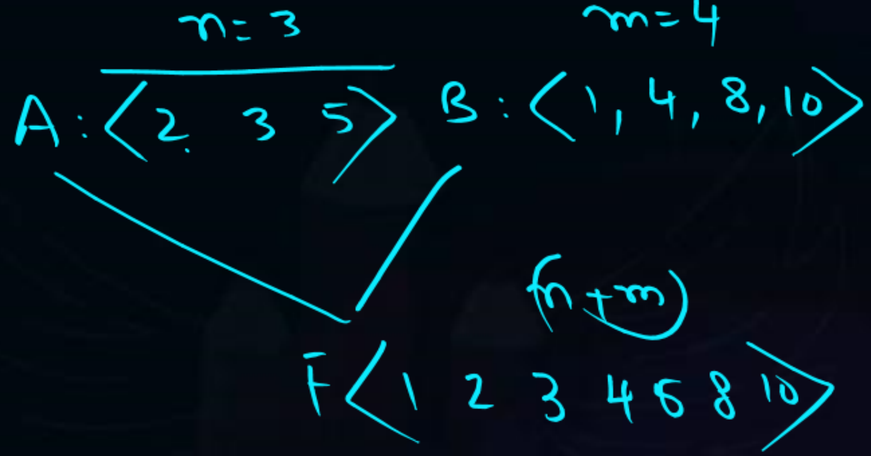
Spanning Trees

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;

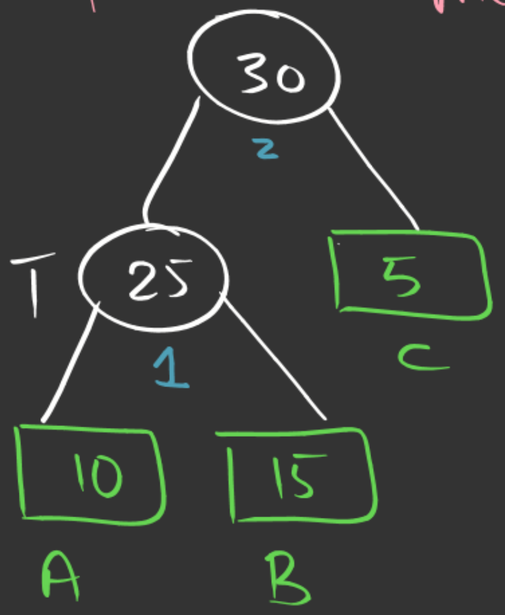
→ Given 'n'-files, it is req'd to Merge them using 2-way Merging, such that, the total no. of record Movements is Minimum;



A: 10 ; B: 15 ; C: 5

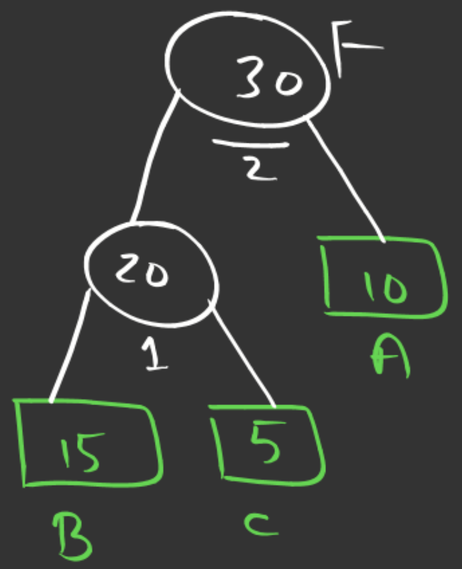
For n-files, it requires (n-1) Merge opns

(2-Way Binary Merge Tree)



: 25 + 30 : 55

⟨A B C⟩



: 20 + 30 : 50

⟨B C A⟩

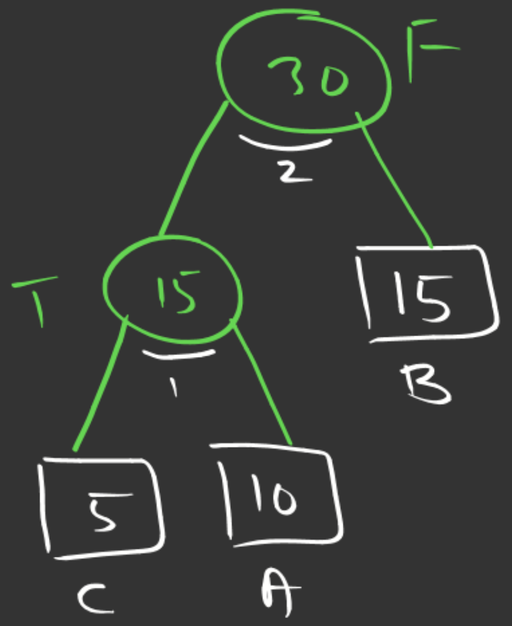
→ We want the pattern which is generating min no. of record Movements

→ Soln Space: n!

Greedy Strategy:

A : 10 ; B : 15 ; C : 5

F1 F2 F3



Total record Mov's : 15 + 30 = 45

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

Wtd. Ext. Path length = $\sum_{i=1}^{n} di * qi$

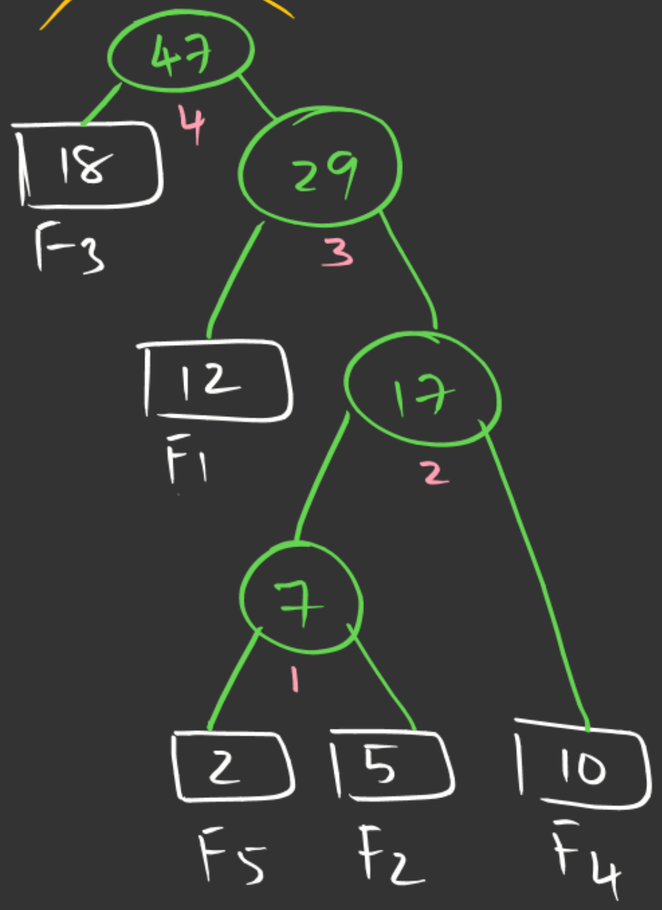
di = distance from root to Fi

qi = Freq / Size of Fi ;

$= 2*10 + 1*15 + 2*5 =$

$= 20 + 15 + 10 = 45$ //

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



Optimal 2-way Binary Merge Tree

No. of record Mov's:

$7 + 17 + 29 + 47 = 100$

Wtd. Ext. Path Length $= \sum_{i=1}^{n} d_i * v_i$

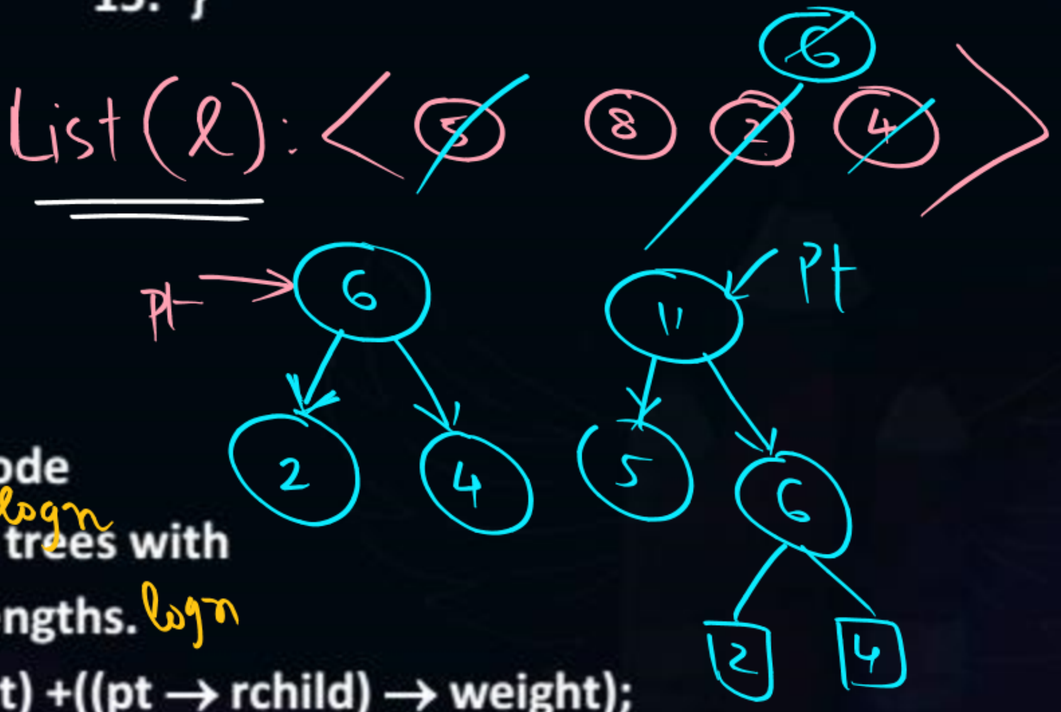
$= 2*12 + 4*5 + 1*18 + 3*10 + 4*2$

$= 100$

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
6.    {
7.    pt := new treenode;// Get a new tree node
8.    (pt→ lchild) := Least(list); // Merge two trees with
9.    (pt → rchild) := Least(list); // smallest lengths.
10.   (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-files :

→ Impl: of list

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

2) Space Complexity

Input : 'n'

Space Complexity : $O(n)$

Nodes

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

↳ (Data Encoding + Data Compression Technique)

↓

(repr. an element with some code)

→ 4 characters/Elements

→ Strait forward / Normal Encoding : 2 bits

(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

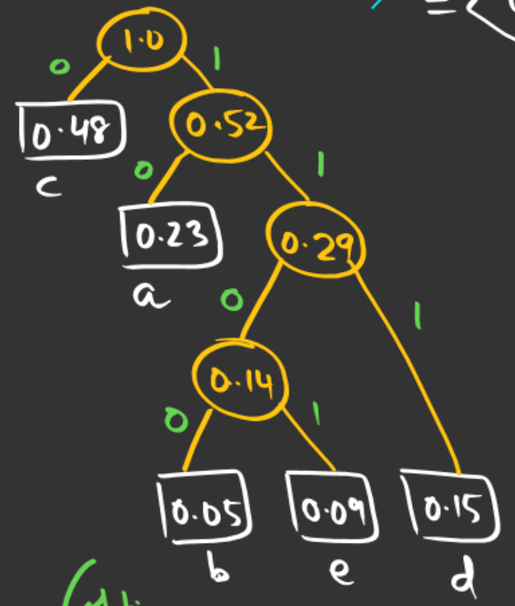
→ Codes are assigned based on frequency; [Non-uniform encoding]

→ Huffman Coding

→ Uniform encoding : 3 bits/char.

$L = \Sigma(a, b, c, d, e)$

$= \langle 0.23; 0.05; 0.48; 0.15; 0.09 \rangle$ ⇒ (opt Merge Pattern Also)



(optimal 2-way Binary encode Tree)

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

: 10010001010011111...

: acaccaaccd...

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

$= 2*0.23 + 4*0.05 + 1*0.48 + 3*0.15 + 4*0.09 = 1.95 \text{ bits/char.}$

Text : <c c a c c c a b d c> IP

(i) Uniform encoding : $10 * 3 = 30$ bits

(ii) Huff. Coding :

a a ccc

<1 0 1 0 0 0 0 1 0 1 1 0 0 1 1 1 0> : 17 bits

ccaccca

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

(b) ecgdf

(c) dchfg

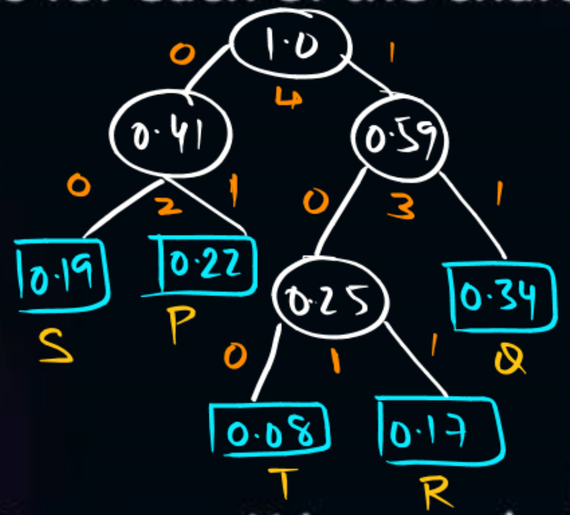
(d) fehdg

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



P → 01 (2) {0, 1, 01}
Q → 11 (2) {1, 1, 11}
R → 101 (3)
S → 00 (2)
T → 100 (3) {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 (225) ✓

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

Topic : III. Greedy Method

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

H.C

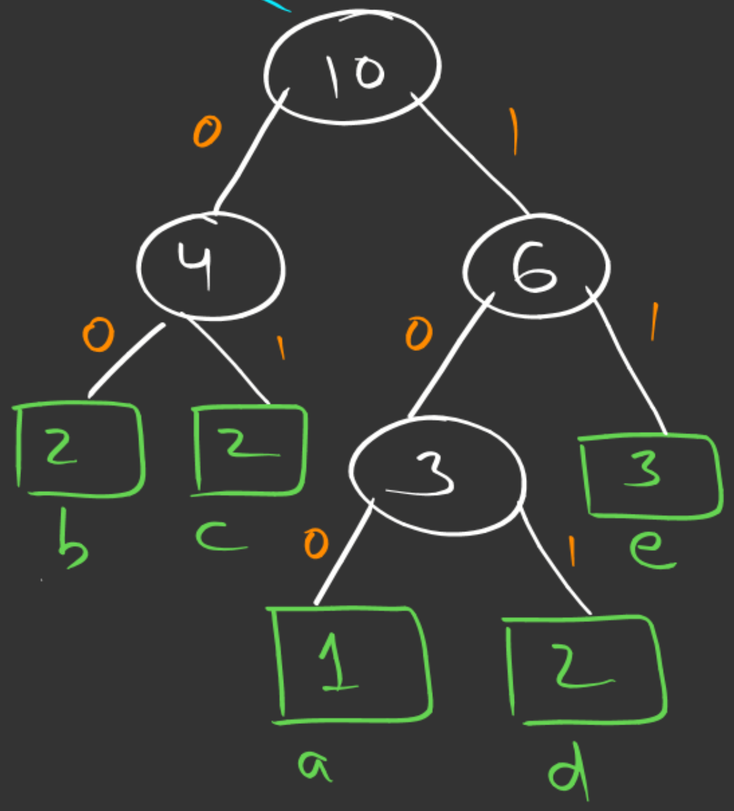
1. For any two letters, the code assigned to one letter must not be a prefix of the code assigned to the other letter.
2. For any two letters of the same frequency, the letter which occurs earlier in the dictionary order is assigned a code whose length is at most the length of the code assigned to the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



a = 1
b = 2
c = 2
d = 2
e = 3

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

Total: $(3+2*2+2*2+3*2+2*3) = 23$

5) Minimum Cost Spanning Tree (MCST)

(Graph based Problem)

Graph (Non-linear D.S)

→ G = (V, E)

- V → Set of vertices
- E → Set of Edges

|V| = n (vertices)

|E| = e (Edges)

- Undirected / Directed
- weighted / Unweighted

Representation of Graphs:

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

THANK - YOU