CS & IT ENGINEERING

Algorithms

Greedy Method

Lecture No.- 04

By- Aditya Jain sir

Topics to be Covered

Topic

Topic OMP

Topic Huffman Encoding

About Aditya Jain sir

1. Appeared for GATE during BTech and secured AIR 60 in GATE in very first attempt - City topper
2. Represented college as the first Google DSC Ambassador.
3. The only student from the batch to secure an internship at Amazon. (9+ CGPA)
4. Had offer from IIT Bombay and IISc Bangalore to join the Masters program
5. Joined IIT Bombay for my 2 year Masters program, specialization in Data Science
6. Published multiple research papers in well known conferences along with the team
7. Received the prestigious excellence in Research award from IIT Bombay for my Masters thesis
8. Completed my Masters with an overall GPA of 9.36/10
9. Joined Dream11 as a Data Scientist
10. Have mentored 12,000+ students & working professions in field of Data Science and Analytics
11. Have been mentoring & teaching GATE aspirants to secure a great rank in limited time
12. Have got around 27.5K followers on Linkedin where I share my insights and guide students and professionals.

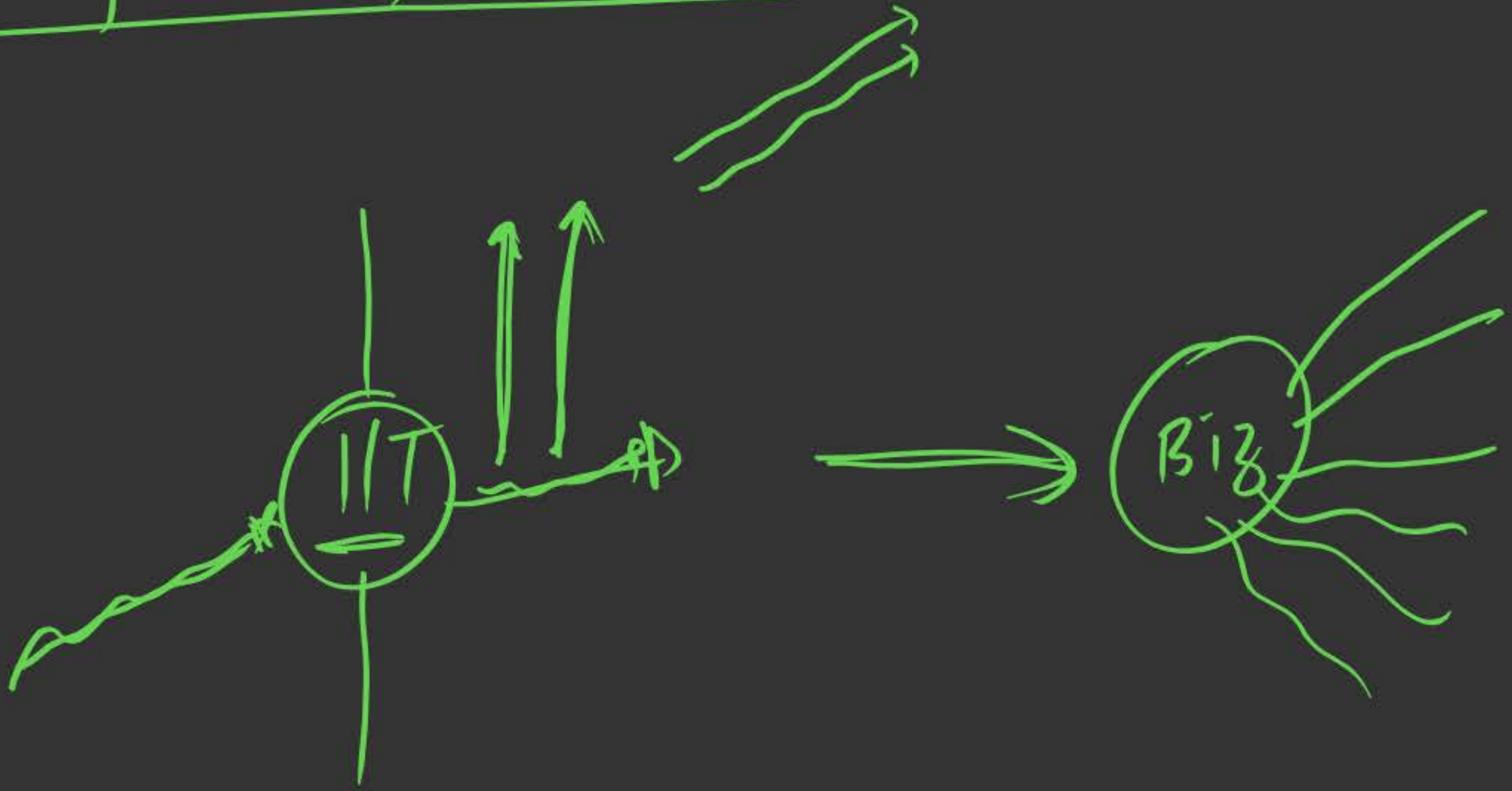
Telegram Link for Aditya Jain sir: https://t.me/AdityaSir_PW

TTTe

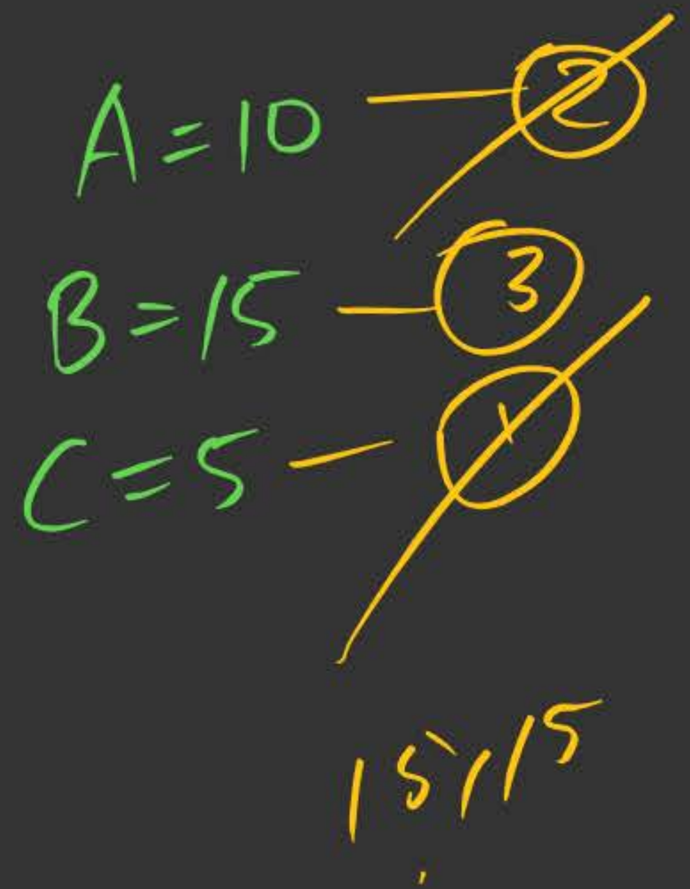
OMP

Brute Force: $\underline{O(n!) \approx O(n^n)}$

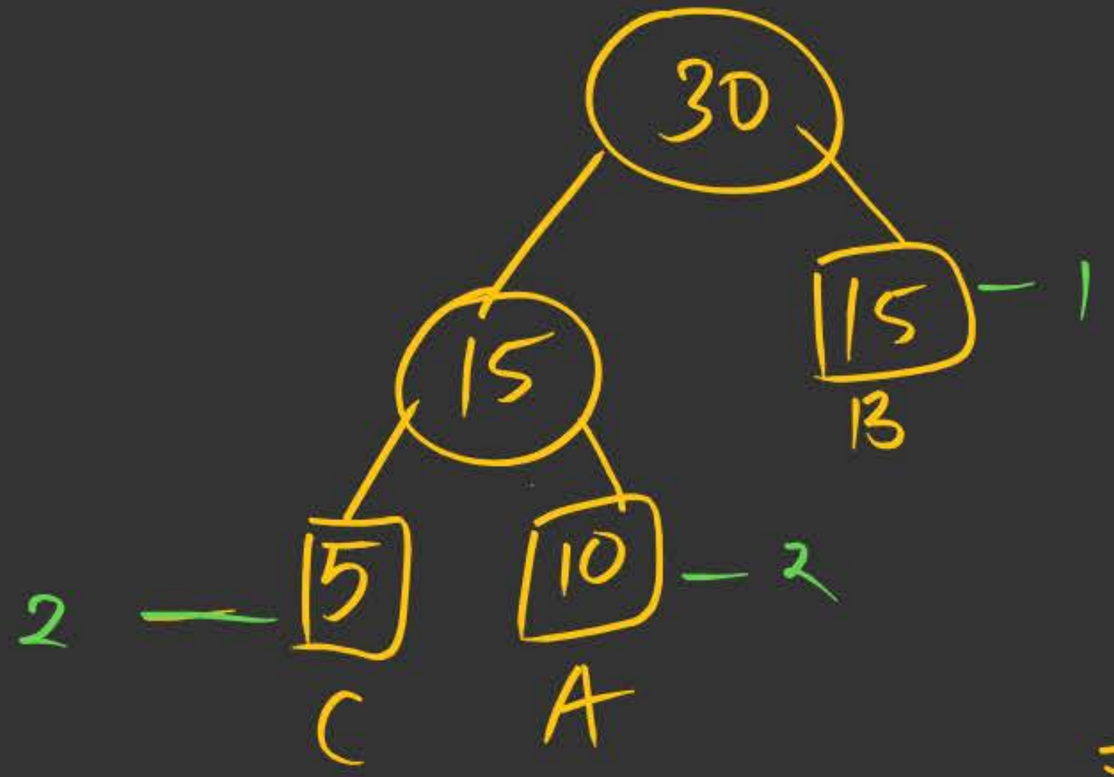
Optimal (Greedy based) Approach :



Optimal (Greedy based) Approach:



Optimal Binary Merge Tree



TRM

= Sum of all internal nodes

$= 15+30 = \boxed{45}$

Appr 2 :-

$$TRM = \sum_{i=1}^{n} d_i q_i = 2 \times 5 + 2 \times 10 + 1 \times 15$$

$$= 10 + 20 + 15$$

$$= \boxed{45}$$

d_i - depth (no. of edges)

q_i - Size

Topic : Greedy of Algorithms

Optimal merge pattern

Eg.1. A : 10 ……(2)

B : 15 ……(3)

C : 5 ……(1)

2. The total number of record movement = weighted external path length of the 2 – way binary merge tree.

Minimum record movement

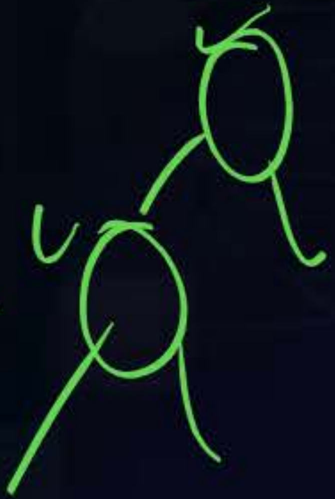
$= \Sigma di * qi$

$= 2\times10 + 1\times15 + 2 \times 5$

$= 20 + 15 + 10$

$= 45$

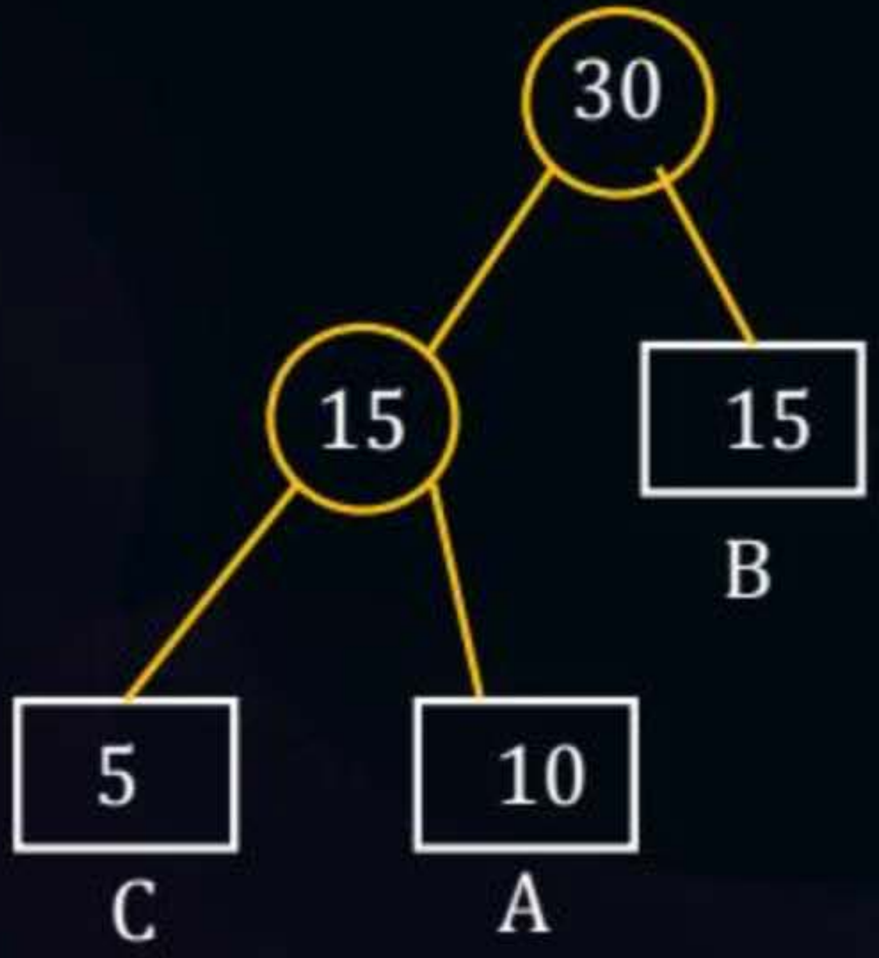
ABC	ACB
BCA	



Topic : Greedy Algorithms

Optimal 2-way Binary merge Tree

Min total record movement = 15 + 30 = 45



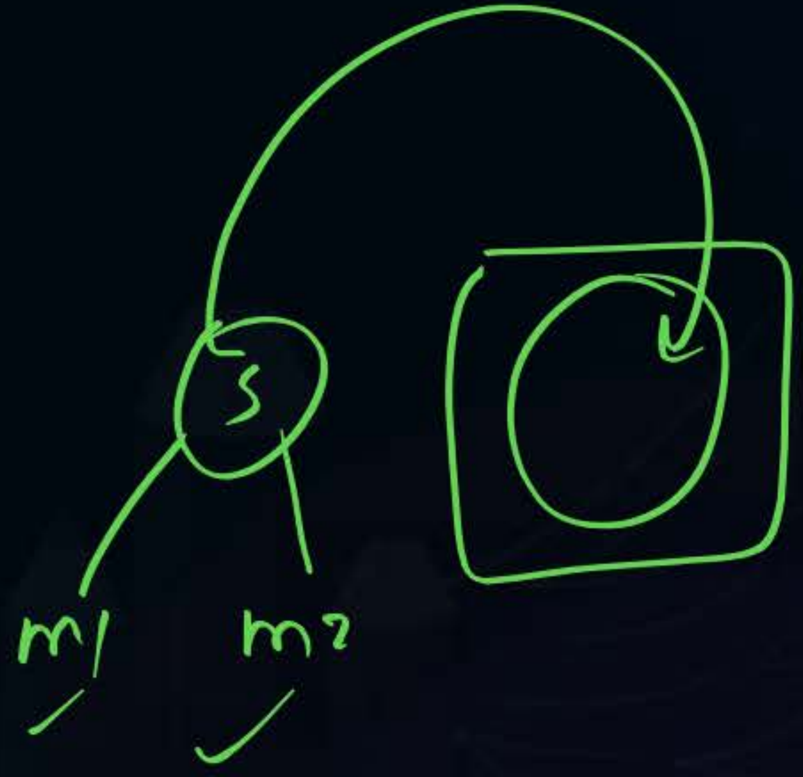
Topic : Greedy Algorithms

Greedy algo for optimal merge patterns

```
Struct tree_node
{
    tree_node  * Lchild;
    tree _nodes * Rchild ;
    int  weight;
};
```

Topic : Greedy Algorithms

```
Algo Tree (n)
{
        // list is a global list of n single nodes binary tree, as described before
        for i : 1 to (n-1) ;→ O(n)
        {
              1.  Pt = new (tree_node);
              2.  Pt → Lchild = minimum (list);
              3.  Pt→ Rchild = minimum (list);
              4.  Pt → weight = ((Pt→ Lchild) → weight)
                                  ((Pt → Rchild) → weight)
              5.  insert (list, Pt);
        }
        return minimum(list);
}
```



Topic : Greedy Algorithms

Time complexity: (for ‘n’ files)

1. Linear linked list:

   (‘n’ elements)

   Insert → $O(1)$

   Minimum → $O(n)$

   Overall TC→ $O(n*n) = O(n^2)$

Topic : Greedy Algorithms

Implementation of list

2. Heap (non-linear)

('n' elements)

Insert ✓

Delete ✓

Dec./Inc. ✓

$\left.\begin{array}{l}\text{Insert} \\ \text{Delete} \\ \text{Dec./Inc.}\end{array}\right\} O(\log n)$

Min Heap: Minimum → O(1)

Overall TC: O(n*logn)

BF Opt

$n! \longrightarrow n\log n$

min TRM?

(Q)

A – 10 — 3 ✓

B – 12 — 4 ✓

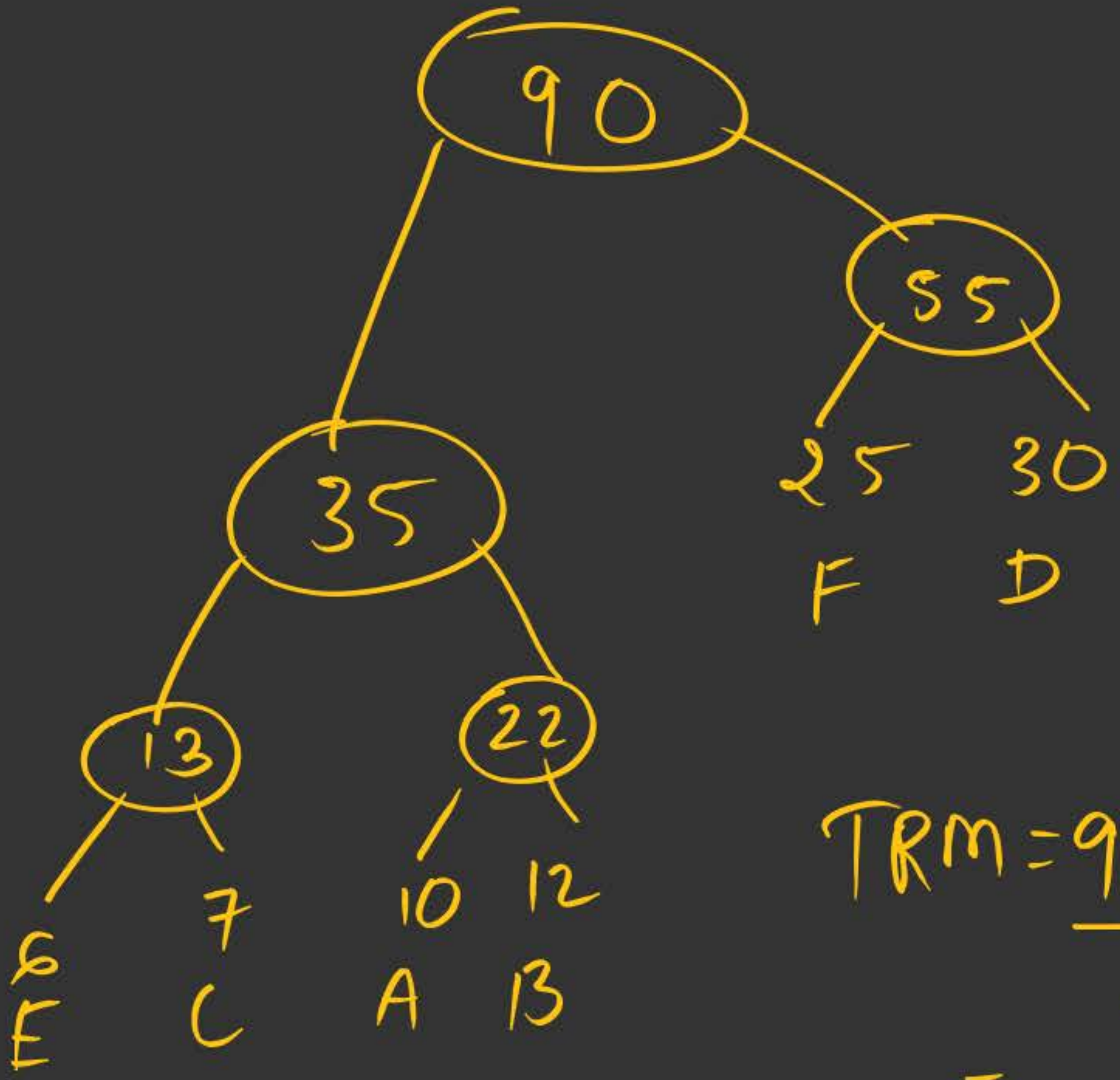
C – 7 — 2 ✓

D – 30 — 6 ✓

E – 6 — 1 ✓

F – 25 — 5 ✓

~~13~~, ~~22~~, 35, 55



22%

$$TRM = 90 + 55 + 35 + 13 + 22$$

$$= 180 + 35$$

$$= \boxed{215}$$

Topic : Greedy Algorithms

3. Huffman Encoding

- Application of optimal merge pattern
- Data encoding and data compression Technique
- Representing some elements with a unique code
- Compression → Reduce size of data without data loss

abc → 123

Topic : Greedy Algorithms

Eg.1. 4 Characters /elements

w, x, y, z

Straight forward Binary encoding

Normal Encoding
or
Uniform Encoding

W → 0 0

X → 0 1

Y → 1 0

Z → 1 1

4

Topic : Greedy Algorithms

If we use uniform encoding:

1. 6 elements → 3 bits
2. 10 elements → 4 bit → $\log_2 10 \rightarrow \lceil 3.x \rceil$

   $\rightarrow 4$

   4 bits → max $2^4 = 16$ elements can be uniquely encoded
3. 17 element → 5 bits……….
4. 100 elements → 7 bits/ elements or characters.

$\boxed{10 - \boxed{16}} \rightarrow 4$

$2 \rightarrow 4$

$3 \leftarrow 6$

6 elem

$\boxed{n \rightarrow 2^n}$

$3 \rightarrow \boxed{8}$

Topic : Greedy Algorithms

In general

To represent N elements using uniform encoding
we need → $\lceil \log_2 (N) \rceil$ bit/character.

$\lceil \log_2 10 \rceil = 4$ $\lceil \log_2 8 \rceil = 3$

Imp. Observations:

1. Using uniform encoding, data compression is not possible.
2. Hence, for data compression we need non-uniform encoding.

Topic : Greedy Algorithms

V.V.imp

Huffman Encoding (non-uniform encoding)

Idea:- Encoding/code → assigned based on the frequency of that character or elements.

Logic:- $(\text{no. of bits to encode}) \propto \frac{1}{(\text{freq.of character})}$

1. Assign more bits to low frequently occurring elements.
2. Assign less bits to high frequently occurring elements.
3. No 2 elements/characters will have the same encoding (as it needs to be unique).

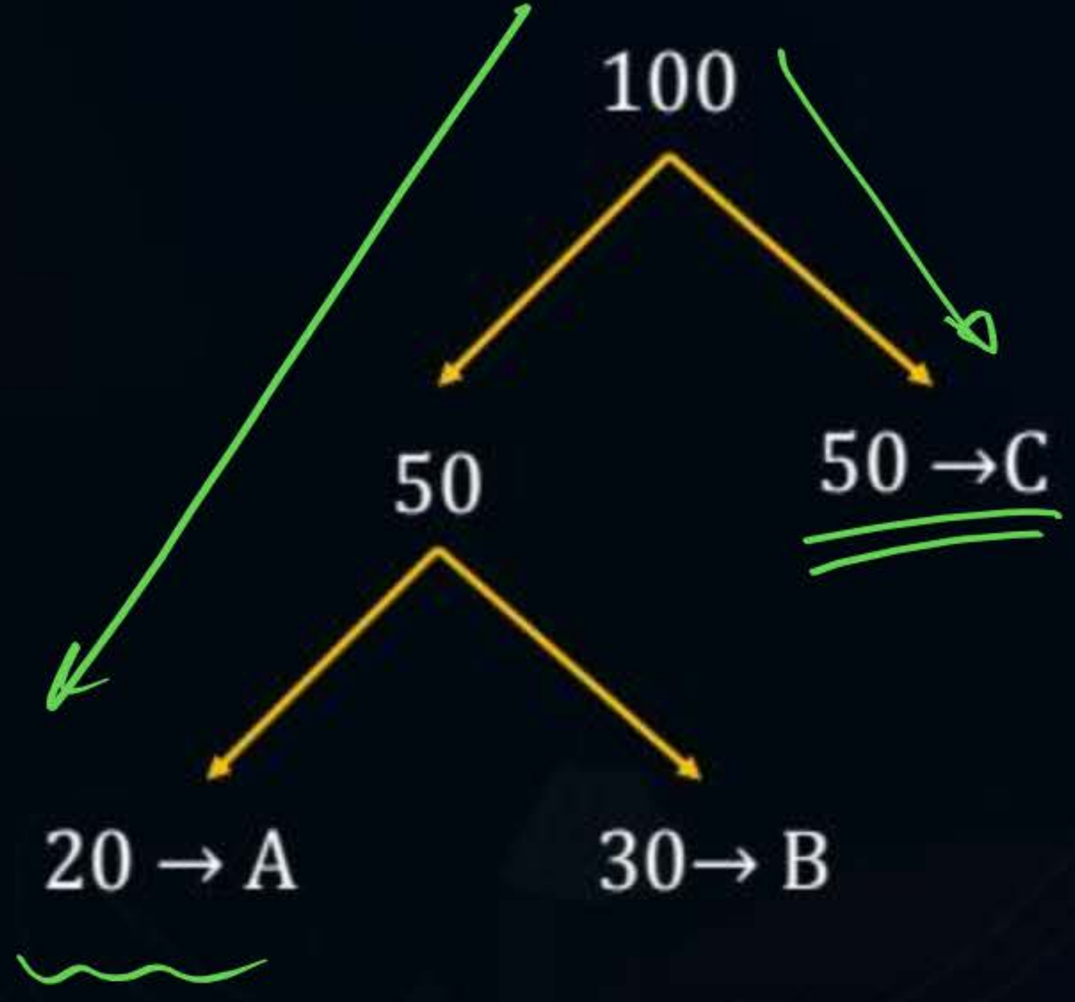
Topic : Greedy Algorithms

Eg.1. A→ 20

B → 30

C → 50

“A B C B C C ....”



Overall total bits compression

Topic : Greedy Algorithms

Note:- Start from lowest level.

Char freq/prob

A → 0.23.............(4)

B → 0.05.............(1)

C → 0.48.............(5)

D → 0.15.............(3)

E → 0.09.............(2)

Avg bits/char

n=5

$$\left\lceil \log_2 5 \right\rceil = 3$$

Priority (low to high)

Topic : Greedy Algorithms

Eg..

Given: Language/ Elements $\Rightarrow \{A,B,C,D,E\} \rightarrow 5$ character

If uniform encoding: $\lceil \log_2 N \rceil$ bits/character

$= \lceil \log_2 5 \rceil$

$= 3$ bit / character

Topic : Greedy Algorithms

2-way Optimal Binary merge tree:

Left Branch → 0

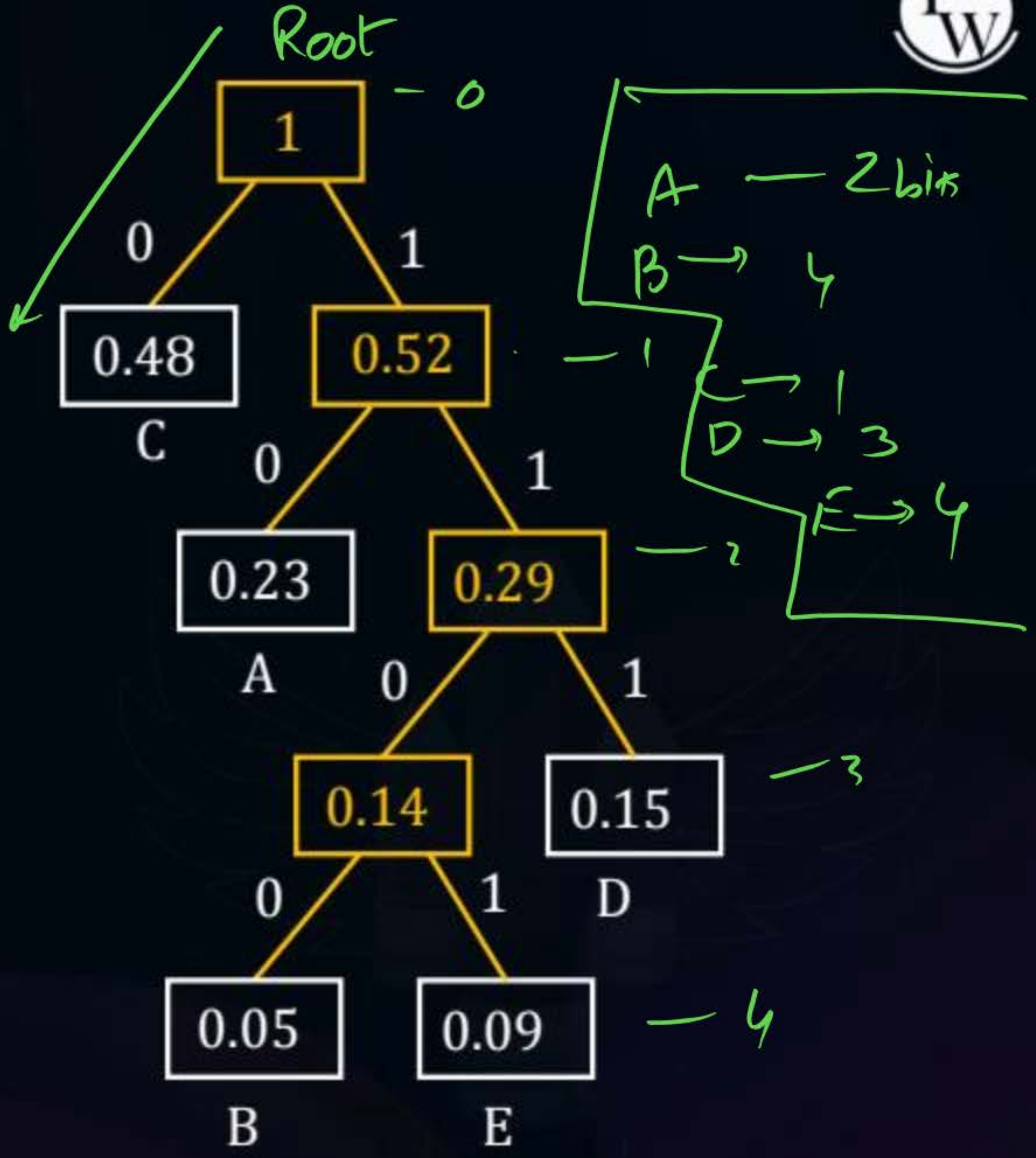
Right Branch → 1

Standard

Left → Smaller

Right→ Larger

A
B
C
D
E



Topic : Greedy Algorithms

A → 10 → 2 bits

B → 1100 → 4 bits

C → 0 → 1 bits

D → 111 → 3 bits

E → 1101 → 4 bits

Non- Uniform encoding & all are unique

Topic : Greedy Algorithms

#Q. Average number of bit required using Huffman encoding?

= Weighted external path length: (Sum of internal nodes)

$= \underline{\sum_{i=1}^{n}} di * qi$

$= \underline{\underline{1.95}}$ bits/char

Topic : Greedy Algorithms

(extension of prev)

#Q. "C C A C C C A B D C"

How many bit required to represent the above text :-

1. Uniform Encoding: → $3 \times 10 = 30$ bits

2. Huffman Encoding: (Non-uniform) → 17 bits

Topic : Greedy Algorithms

2. Using Huffman Encoding:

Text:	C	C	A	C	C	C	A	B	D	C
	0	0	10	0	0	0	10	1100	111	0

Final Encoding:

0 0 1 0 0 0 0 1 0 1 1 0 0 1 1 1 0

Total = 17 bit compressions

A → 2

B → 1

C → 6

D → 1

E → 0

Total bit

$= 2 \times 2 + 1 \times 4$

$+ 6 \times 1 + 1 \times 3$

$+ 0 \times 4$

$= 17$ bits

2 mins Summary

Topic

Topic omp

Topic HE

THANK - YOU