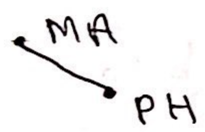
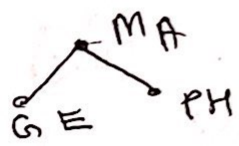
APPLICATIONS OF TREES

Binary Search Trees

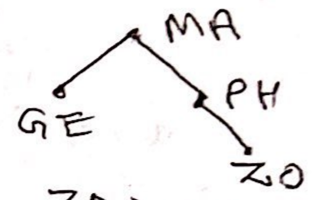
Example Form a binary search tree for the words mathematics (MA), PHYSICS (PH), geography (GE), Zoology (ZO), ~~meto~~ meteorology (ME), geology (GEO), Psychology ~~(PHY)~~ (PSY) & chemistry (CH), using alphabetic order.



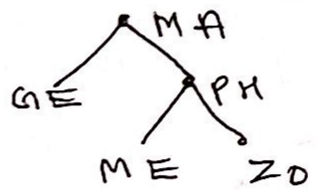
PH > MA



GE < MA

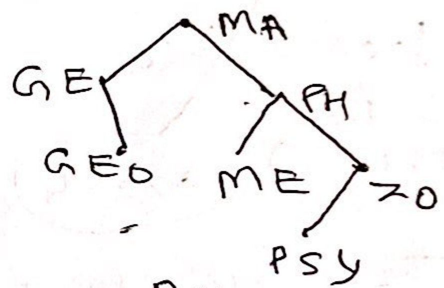


ZO > MA
ZO > PH

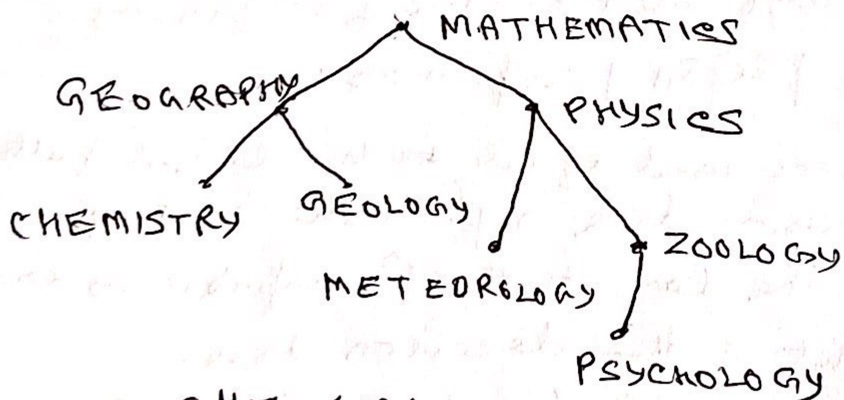


ME > MA
ME < PH

GEO < MA
GEO > GE



PSY > MA
PSY > PH
PSY < ZO



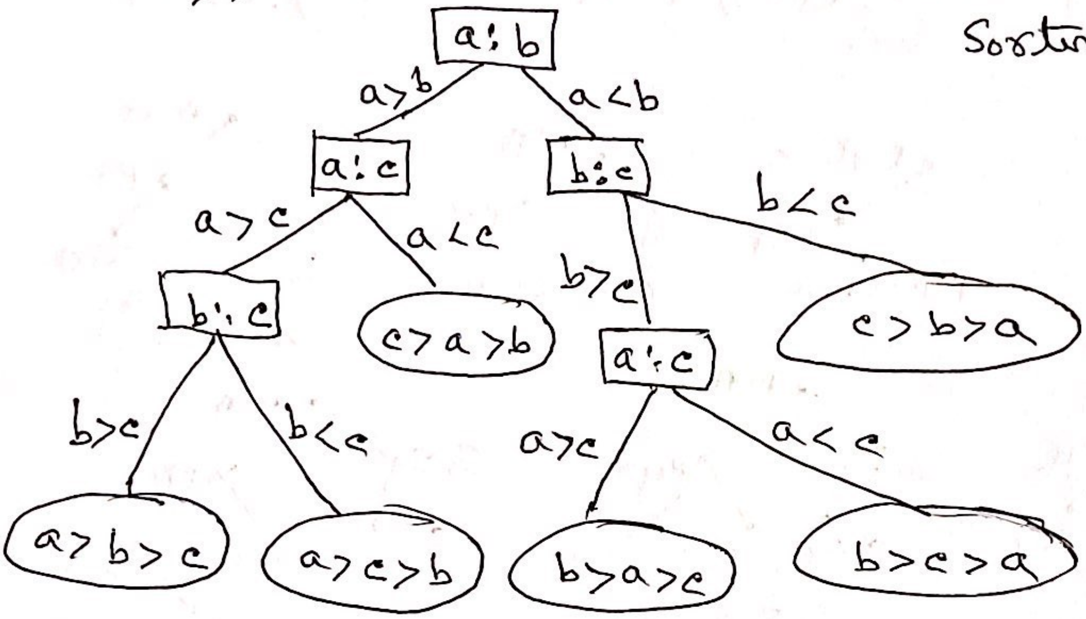
CHE < MA
CHE < GE

$\boxed{\text{Decision Trees}}$ Rooted trees → series of decisions

A rooted tree in which each internal vertex corresponds to a decision with a subtree at these vertices for each possible outcome of the decision. The possible solns of the problem correspond to the paths to the leaves of the rooted tree.

— A decision tree for ordering the elements of the list a, b, c

Sorting a, b & c



— The complexity of sorting algo.

$\boxed{\text{Theorem}}$ A sorting algo based on binary comparisons requires at least $\lceil \log n! \rceil$ comparisons.

D The most comparisons used equal ~~to~~ the longest path length in the decision tree rep^{ing} ~~the~~ the sorting procedure. ⇒ The largest # of comparisons ever needed = height of the decision tree.

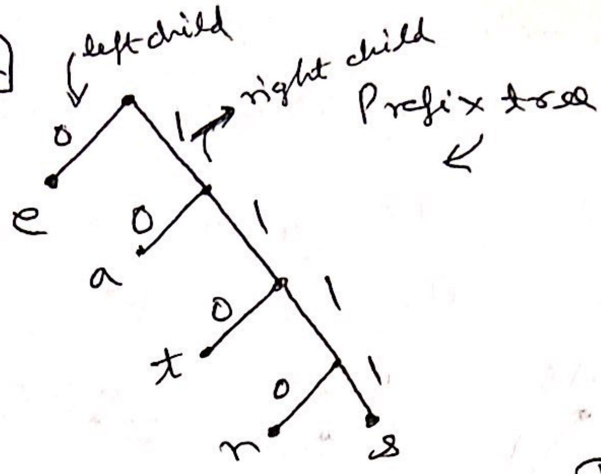
(Now the height of a binary tree with $n!$ leaves is $\geq \lceil \log n! \rceil$) ⇒ $\geq \lceil \log n! \rceil$ comparisons needed ▨

Now $\lceil \log n! \rceil = \Theta(n \log n)$

Cor ① The # of comparisons used by a sorting algo to sort n elements based on binary comparisons is $\Omega(n \log n)$

② note 12

Prefix Code



$e \rightarrow 0$
$a \rightarrow 10$
$t \rightarrow 110$
$n \rightarrow 1110$
$s \rightarrow 1111$

Decoding

$\underbrace{1111}_{s}\,\underbrace{10}_{a}\,\underbrace{1110}_{n}\,\underbrace{0}_{e}$

ate $\xrightarrow{\text{encoding}}$ 10110

Data Compression (Huffman Coding)

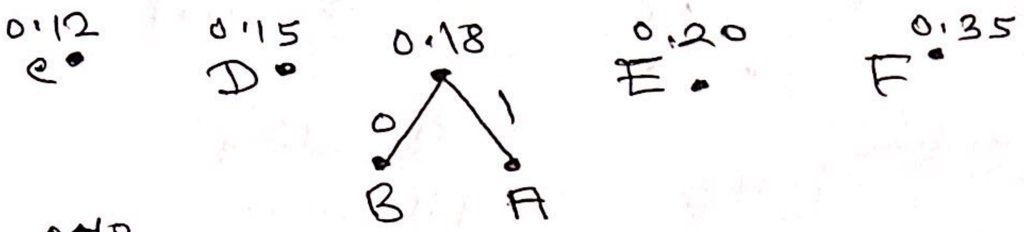
Huffman Coding Algo

Procedure Huffman (C : symbols a_i with freq. w_i $(1 \le i \le n)$)
F := forest of n rooted trees, each consisting of the single vertex a_i and assigned wt. w_i
while F is not a tree
begin
Replace the rooted trees T & T' of least wt. from F with $w(T) \ge w(T')$ with a tree having a new root that has T as its left subtree and T' as its right subtree. Label the new edge to T with 0 and the new edge to T' with 1
Assign $w(T) + w(T')$ as the wt. of the new tree
end

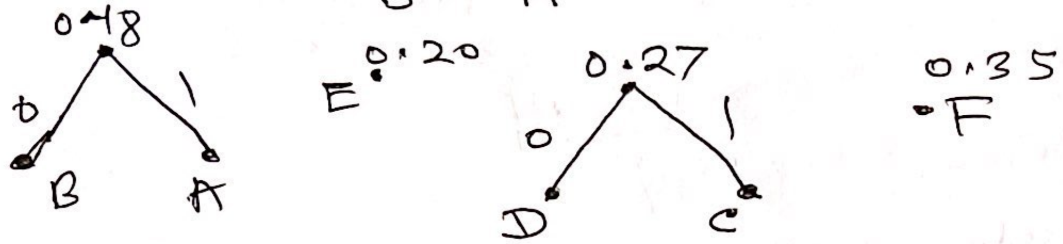
Example Symbols with freq.:

A: 0.08 B: 0.10 C: 0.12 D: 0.15 E: 0.20 F: 0.35

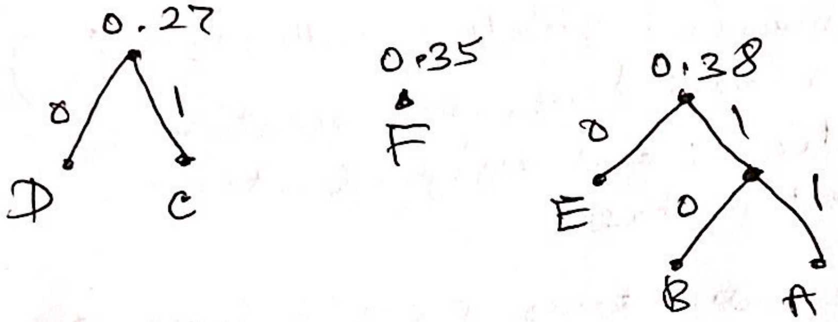
Initial Forest



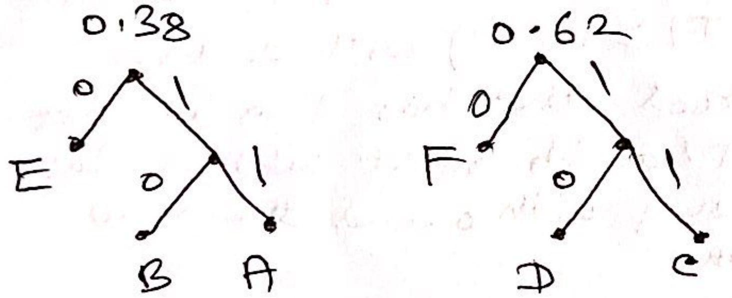
step 1



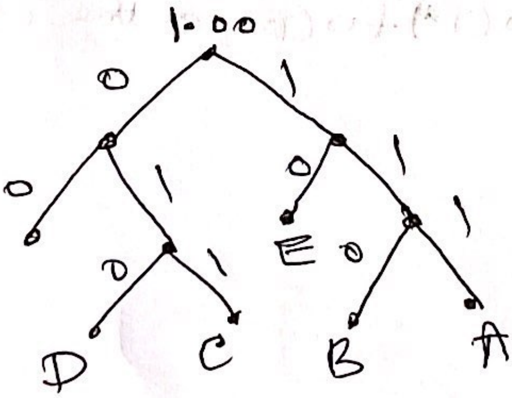
step 2



step 3



step 4



step 5

A → 111 C → 011 E → 10
B → 110 D → 010 F → 00

The avg. no. of bits to encode a symbol using this encoding = 3×.08 + 3×0.10 + 3×0.12 + 3×0.15 + 2×0.20 + 2×0.35 = 2.45 bits

U notes12