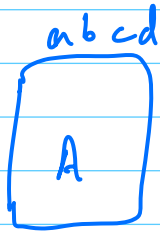
HUFFMAN CODING

Lossless data Compression Algorithm

File size  → Compress (Reduce size)



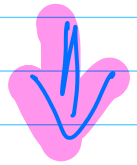
Total char 200

a	0 0
b	0 1
c	1 0
d	1 1

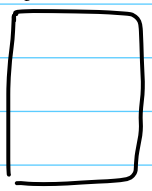
} fixed length code

= 200 x 2 = 400 bits to represent the file A

compress using Huffman code



" Huffman uses Variable length codes for encoding symbols based on Probability of Occurences "



occurrence — give

a = 100 (Small l)
b = 80
c = 10
d = 10 (large l)

} Frequency of occurences

Variable length Coding

① Find frequency of occurence

② Construct a tree

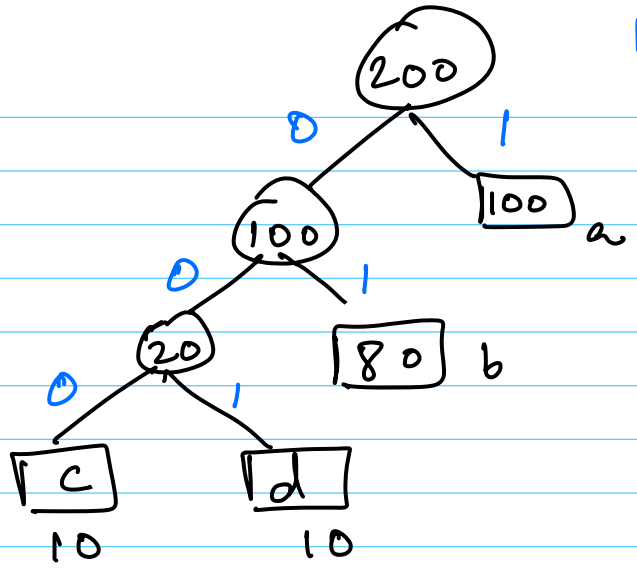
(last nodes = least frequent)

a = 100
b = 80
c = 10
d = 10

200

left = small 0
right = large 1

Start with last node with least freq



a = 100 → 1
b = 80 → 01
c = 10 → 000
d = 10 → 001

(no repetetition of pattern).

Total bits ⇒ 100 + 80×2 + 10×3 + 10×3
= 100 + 160 + 30 + 30
= 100 + 160 + 60
= 100 + 220
= 320 bits

320 bits < 400
with huffmann without huffmann

problem :

Solution

1 2 3 6 9 10 15

g e f c b d a

12, 9, 10, 15

9, 10, 12, 15