

Flow

①

Make freq. dict. using Hash Map

Eg "abodg"

a:2

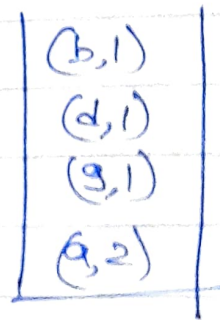
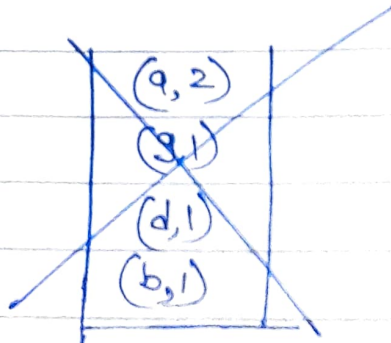
b:1

d:1

g:1 3

②

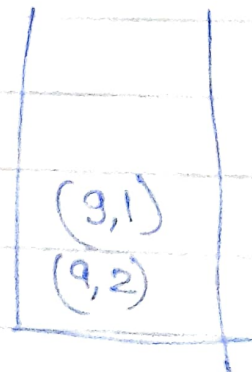
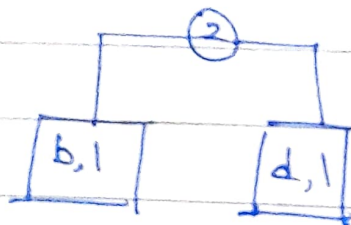
Build ^{Min} Heap using nodes



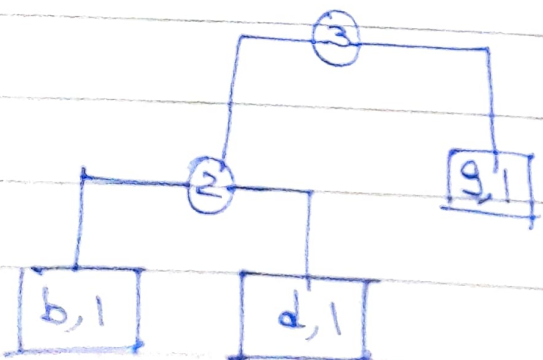
③

Build Tree

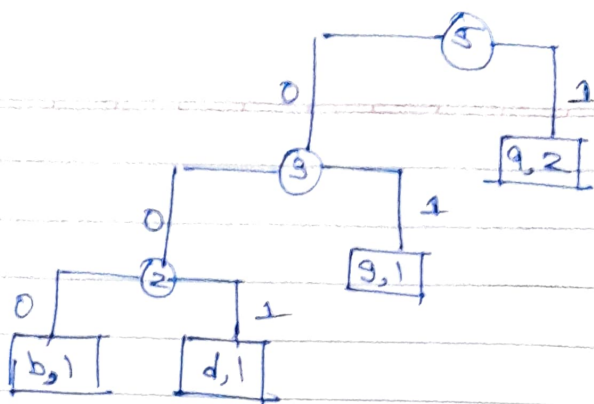
1)



2)



3)



4)

Encode each char

$b \rightarrow 000$, $d \rightarrow 001$, $g \rightarrow 01$, $a \rightarrow 1$

$\therefore \underbrace{abaddg}_{5 \text{ Bytes}} \rightarrow \underbrace{1000100101}_{1 \text{ Byte}} \rightarrow L=10$
 Just Greater than 2 Bytes

Space Red!

Since, "a" \rightarrow had Max. freq, it was popped last, hence got smallest code \rightarrow Less space for Most frequent char

5)

Padding

$$\text{padding_required} = 8 - (L \% 8) = 8 - (10 \% 8) = 6$$

Text \rightarrow 10001001 01000000

padding_info = 00000110

\Rightarrow Text \rightarrow 00000110 10001001 01000000

6)

Convert into Bytes using bytearray C)

Now Text is Compressed

#

Decompress

1. Convert bits to bytes using ord()

Text \rightarrow 00000110
padded_info

extra_padding = 6

padded encoded text

10001001 01000000

encoded_text

Reverse Mapping =

000: b

001: d

01: g

1: a 3

text = 10001001 01

a b a d g \Rightarrow a b a d g

decoded text

Text is Decompressed