Tuesday, 28 December 2021 12:54 PM

Huffman coding

→ lossless data compression algorithm.

→ assign variables - length codes to input characters. length of assigned codes are based on frequency of corresponding character.

→ The most frequent character gets the smallest code and least frequent character gets the longest code.

(A) 0100001, (B) 0100010

A B B C D B C C D A A B B E E E B E A B = 20 char

(A – E) range

① way

ASCII → 127 char

A → (65) → 010001 = 8 bits

Total length of seq = 20 × 8 = 160 bits.

↗ fixed length coding

② way

Range → (A → E) ↳ 5 char

for 4 char,

00 → a
01 → b
10 → c
11 → d

For representing 5 char → 3 bits

0	0	0	→	A
0	0	1	→	B
0	1	0	→	C
0	1	1	→	D
1	0	0	→	E

8 bits

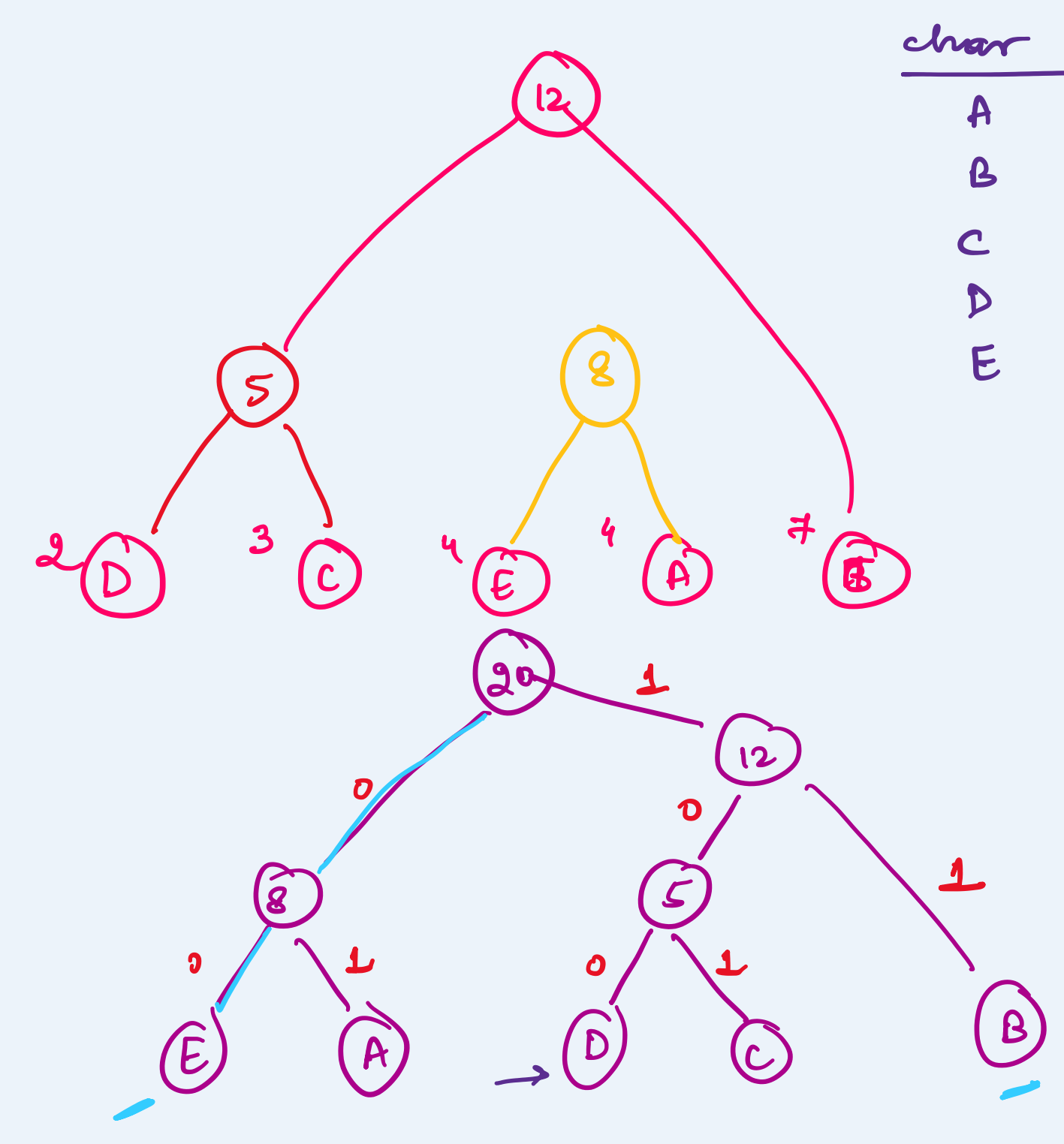
A → 000, B → 001, B → 001, C → 010, D ...

Total bits = 20 × 3
= 60
+
5×3 + 8×5
= 60 + 15 + 40
= 115 bits

③ way

A B B C D B C C D A A B B E E E B E A B

char	freq	
A	4	01
B	7	11
C	3	101
D	2	100
E	4	00



(2, 3, 4, 4, 7)
↓
(5, 4, 4, 7)
↓
(5, 8, 7)
↓
(12, 8)

left → 0
right → 1

Total bits → 4×2 + 7×2 + 3×3 + 2×3 + 4×2
⇒ 8 + 14 + 9 + 6 + 8
= 45 bits
+
5×8 + 12
= 97 bits

TC → O(nlogn)

Prefix codes: No code is prefix of another code.

A → 0
B → 1
C → 01

001 → AC, AAB

→ David Huffman in 1951

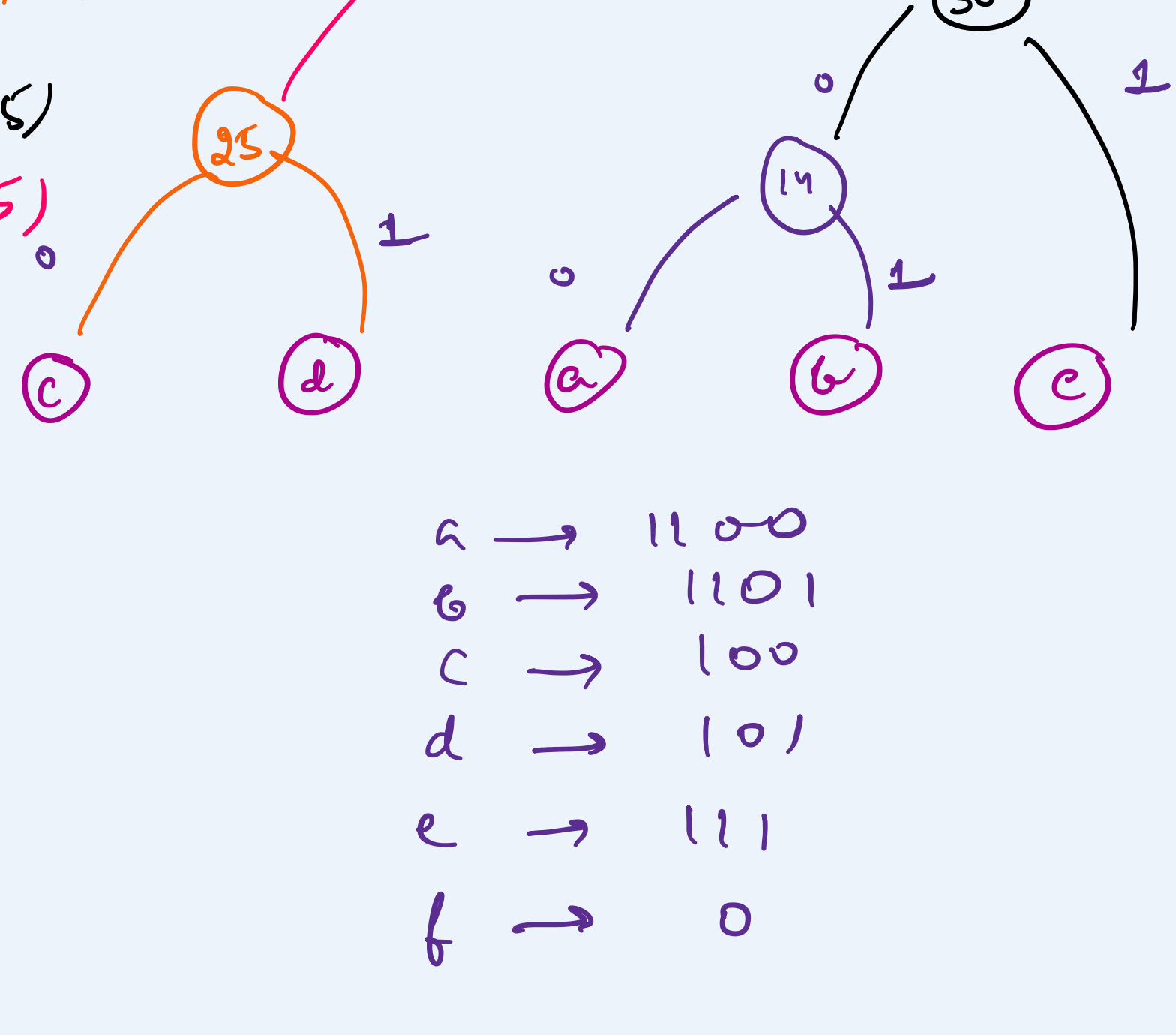
→ encoding follows the prefix rule

→ most generated char will get the small code & least generated char will get the longest code.

For practice:

char	freq
a	5
b	9
c	12
d	13
e	16
f	45

(5, 9, 12, 13, 16, 45)
(12, 13, 14, 16, 45)
(14, 16, 25, 45)
(25, 30, 45)
(45, 55)



a → 1100
b → 1101
c → 100
d → 101
e → 111
f → 0

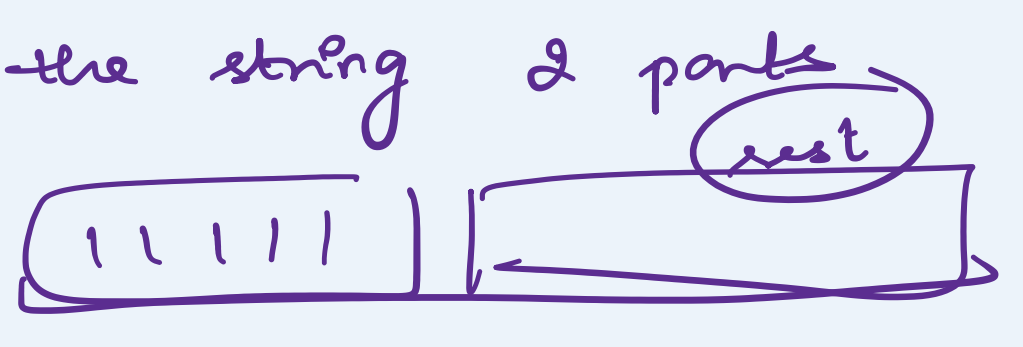
Q. Maximum binary string after change.

000110
↓
000101
↓
100101
↓
110101
↓
110011
↓
110011

00
↓

000 .. 1111
↓
111

Divide the string 2 parts



① for rest part, we can always use 10 → 01 to put all ones to the end of the string and hence move all zeroes ahead of these ones.

② for all zeroes, apply 00 → 10, from left to right, till only one 0 is remaining.