

# Huffman Coding

- lossless data compression algorithm
- assign variable-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 largest code.

$\Rightarrow$

① way

ASCII  $\rightarrow$  127 char  
 A  $\rightarrow$  65  $\rightarrow$  0100001  
 = 8 bits

Total length of seq =  $20 \times 8$   
 = 160 bits.

fixed length coding

② way

for 4 char

0 0  $\rightarrow$  a  
 0 1  $\rightarrow$  b  
 1 0  $\rightarrow$  c  
 1 1  $\rightarrow$  d

(A  $\rightarrow$  E) range

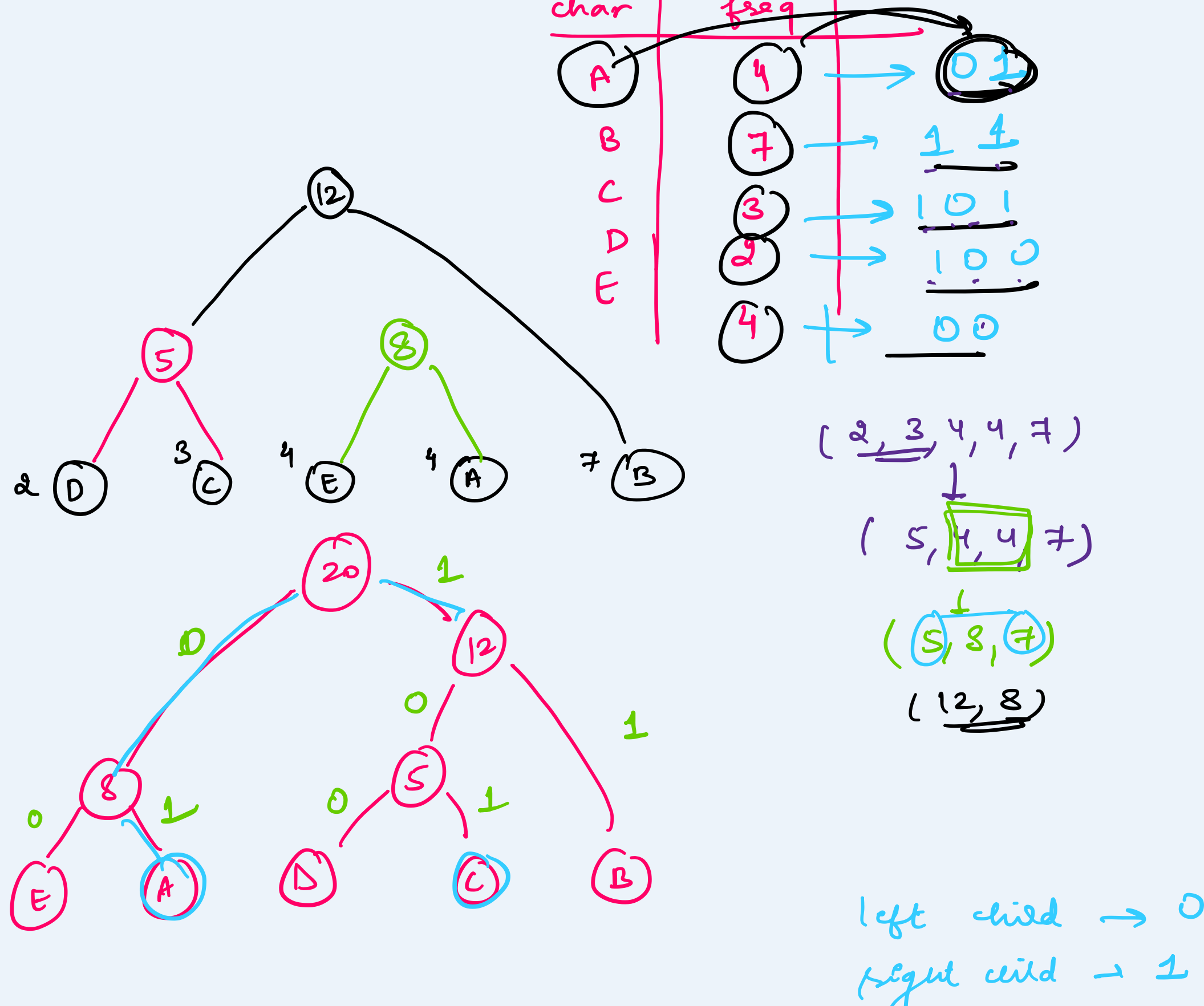
5 char

For representing 5 char  $\rightarrow$  3 bits

Total bits  
 =  $20 \times 3 = 60$   
 +  
 $5 \times 3 + 8 \times 5$   
 =  $60 + 15 + 40$   
 = 115 bits

③ way

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



Total bits  $\rightarrow 4 \times 2 + 7 \times 2$   
 +  $3 \times 3 + 2 \times 3$   
 +  $4 \times 2$   
 =  $8 + 14 + 9 + 6 + 8$   
 = 45 bits

$\downarrow$   
 $5 \times 8 + 12$   
 = 97 bits

TC  $\rightarrow (n \log n)$

Prefix codes: No codes is prefix of another code



- David Huffman in 1951
- encoding follows the prefix rule
- most generated char will get the small code & least generated char will get the large code.

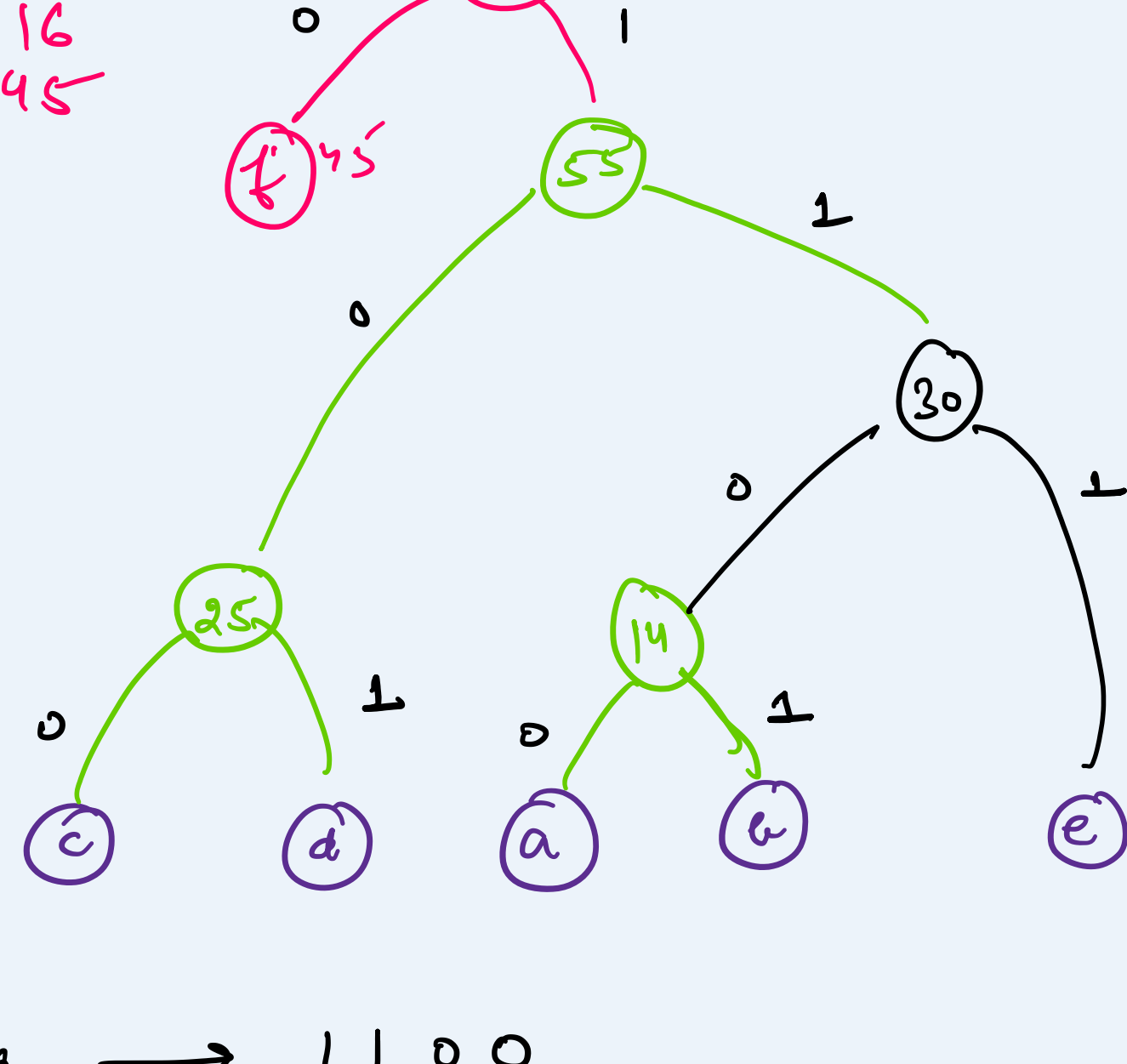
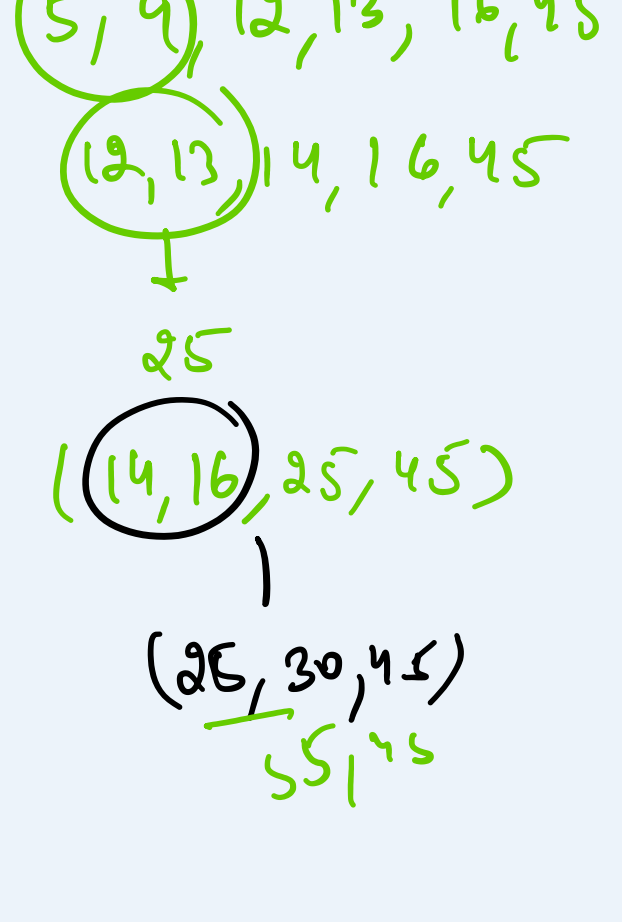
TC  $\rightarrow O(n \log n)$

For practice

char freq

a  
 b  
 c  
 d  
 e  
 f

5  
 9  
 12  
 13  
 16  
 45



a  $\rightarrow$  1100  
 b  $\rightarrow$  1101  
 c  $\rightarrow$  100  
 d  $\rightarrow$  101  
 e  $\rightarrow$  111  
 f  $\rightarrow$  0