Huffman coding ⟶ lossless algorithm
data compression.

string → "aabcdag" ⟶ 7x2 bytes or
7x2x8 = 112 bits.

size of "y/o"

2 hashmaps:

encoder

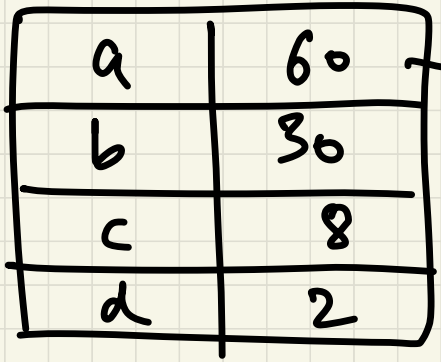
char	string
a	
b	
c	
d	

decoder

string	char
	a
	b
	c
	d

steps:

1. pass the string (data, aka feeder)
2. make frequency map



→ a is coming 60 times

③ For every key in freq map, create a node & insert that node in a min heap / PQ.

Node data: char data; int wt → frequency

a

60

Node left & right

a	60
x	x

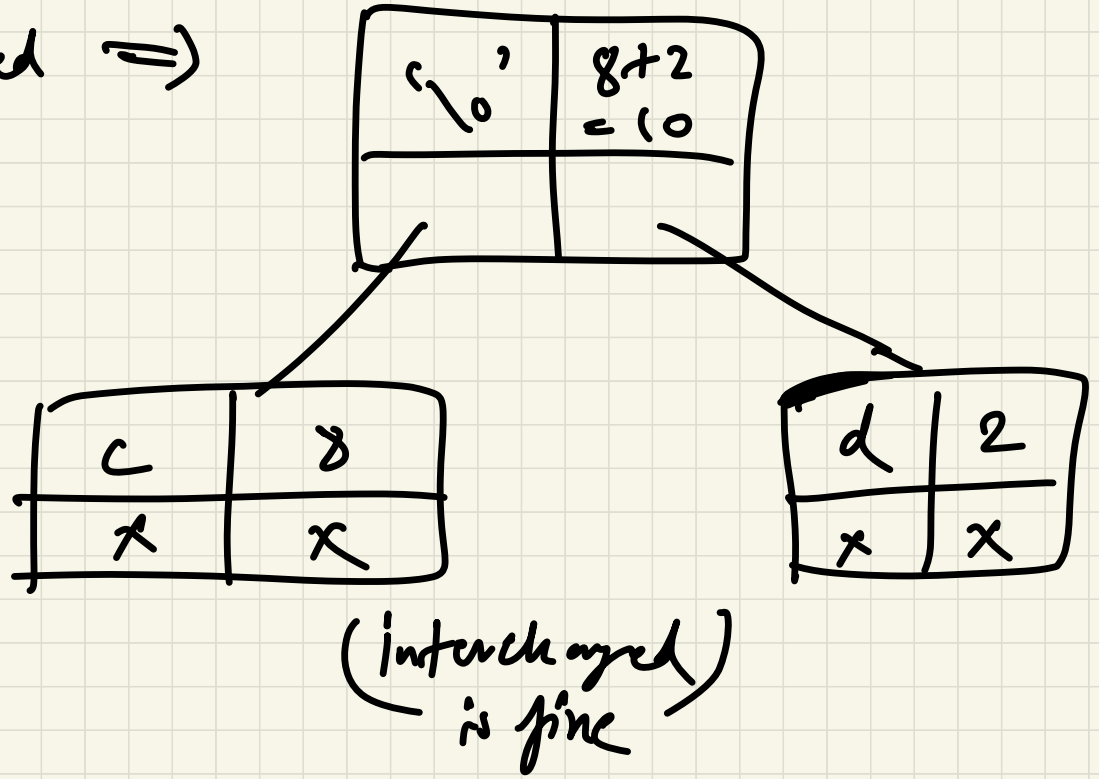
b	30
x	x

c	8
x	x

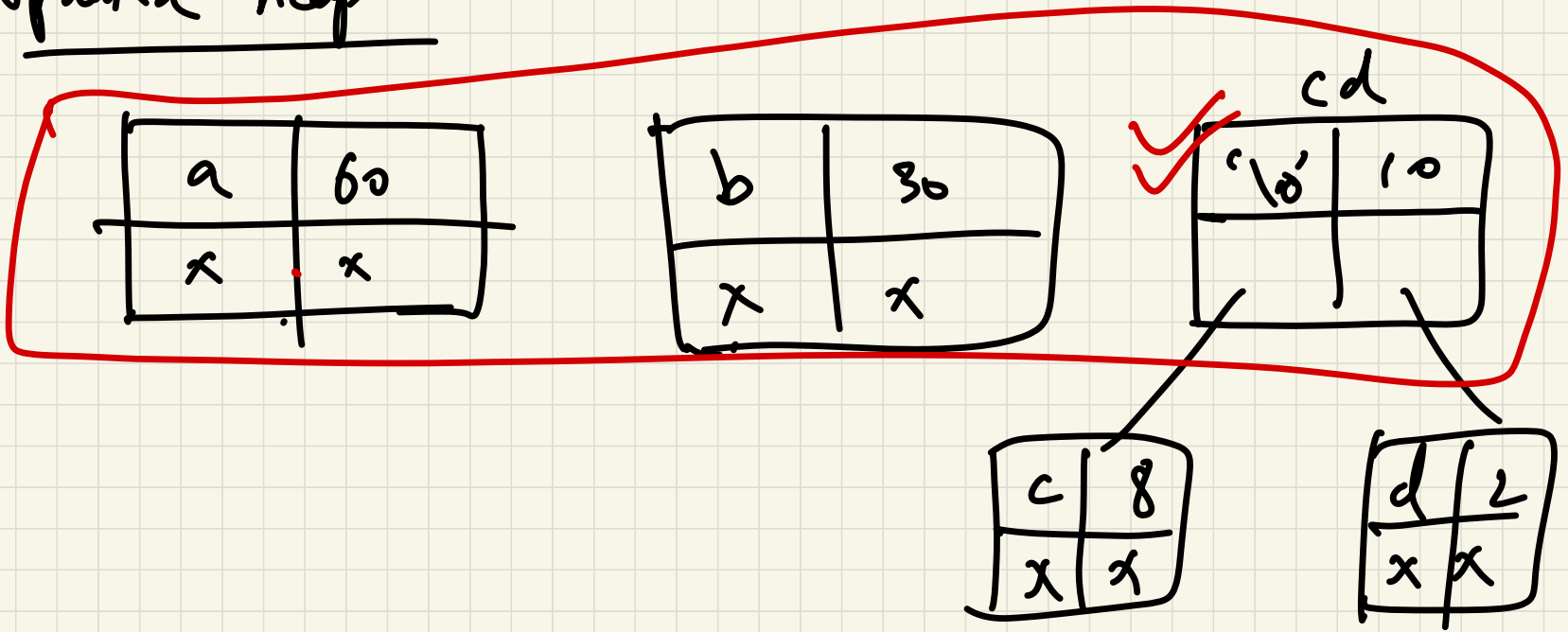
d	2
x	x

④ Remove 2 elements from heap & combine.

c, d removed ⟹

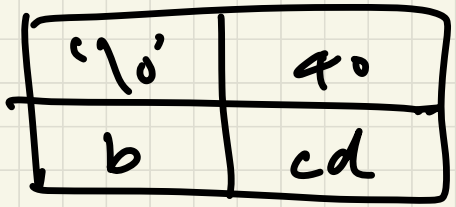


Updated heap:

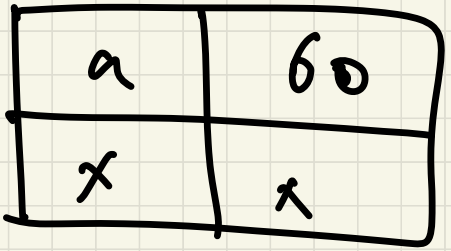


cd & b removed:

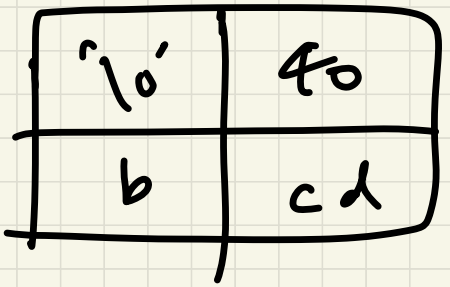
bcd



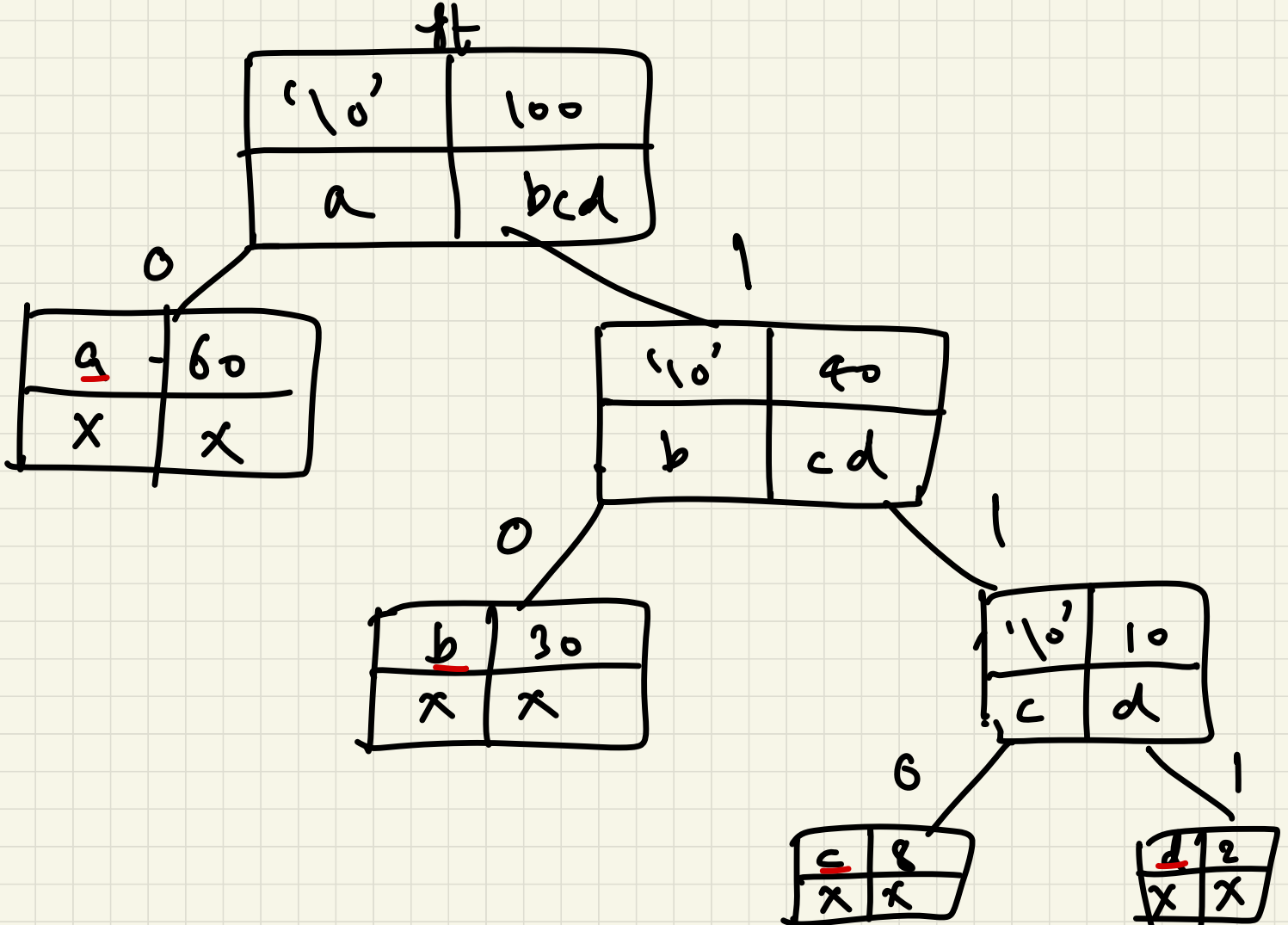
Updated heap:



bcd



(5)



⑥ encoder

a	0
b	10
c	110
d	111

↗ no valid code is prefix of another value

a → 0

decoder

0	a
10	b
110	c
111	d

⑦ How to encode / decode ?

String = a b b c c d a

= 1 1 2 bits (7 x 2 x 8)

a b b c c d a

0 10 10 110 110 1110 → 15 bits

a b b c c d a

Space = O(N)

Time = 2 * (n-1) times

= O(n log n)