03 March 2025 - Algorithms - Week 10

Maximum Independent Set Problem

Solution: see slides

Claim: $\exists$ a Max Ind. Set that contains leaf V.

Let S be some maximum independent set.

If $v \in S$ ✓

Otherwise:

Case 1: $w \notin S$, then $S \cup \{v\}$ is also independent.

↳ not possibe since S is a max ind. set

Case 2: $w \in S$, then $S \setminus \{w\} \cup \{v\}$ is also max ind. set.

Fractional knapsack

Text Compression : Huffmann Codes

Text encoding

text file : 1000 characters

1 byte → 1 char (ASCII)

1000 chars → 1000 bytes → 8000 bits

But 1 byte can encode 256 characters → we don't need those many characters (52, upper + lower)

6 bits per char ⇒ 6000 bits.

Suppose, in 1000 chars

only A B C D ⟶ (2 bits) 2000 bits

A	:	800
B	:	100
C	:	50
D	:	50

A → 0

B → 10

C → 01

D → 11

Wrong encoding

0 | 1 1 | 0 → Ambiguous

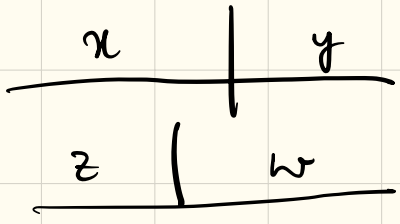
A D A

0 1 | 1 0

C B

A	→	0	800	
B	→	10	+ 200	→ 1300 bits
C	→	110	+ 150	
D	→	111	+ 150	

No string is a prefix of another



Prefix - free encoding

$$xy = zw$$

Prefix - free encoding:

Each letter $s \in \Sigma^*$

$s(c_1)$ is not a prefix of $s(c_2)$ $\forall\ c_1 \neq c_2$

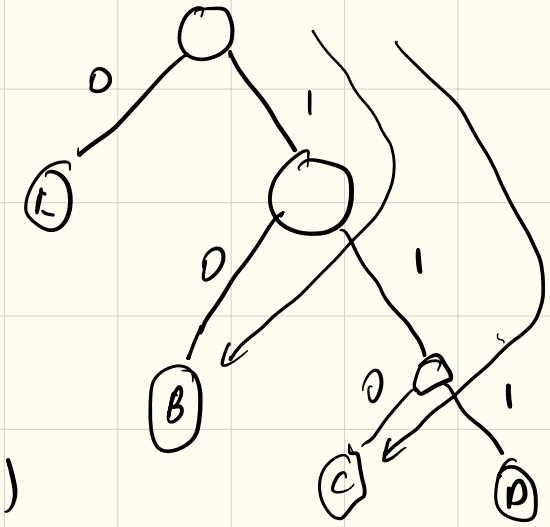
0	1 1 0	0	0	1 1 0	1 1 1
A	C	A	A	C	D

High frequency char :→ less bits

Low frequency char :- more bits.

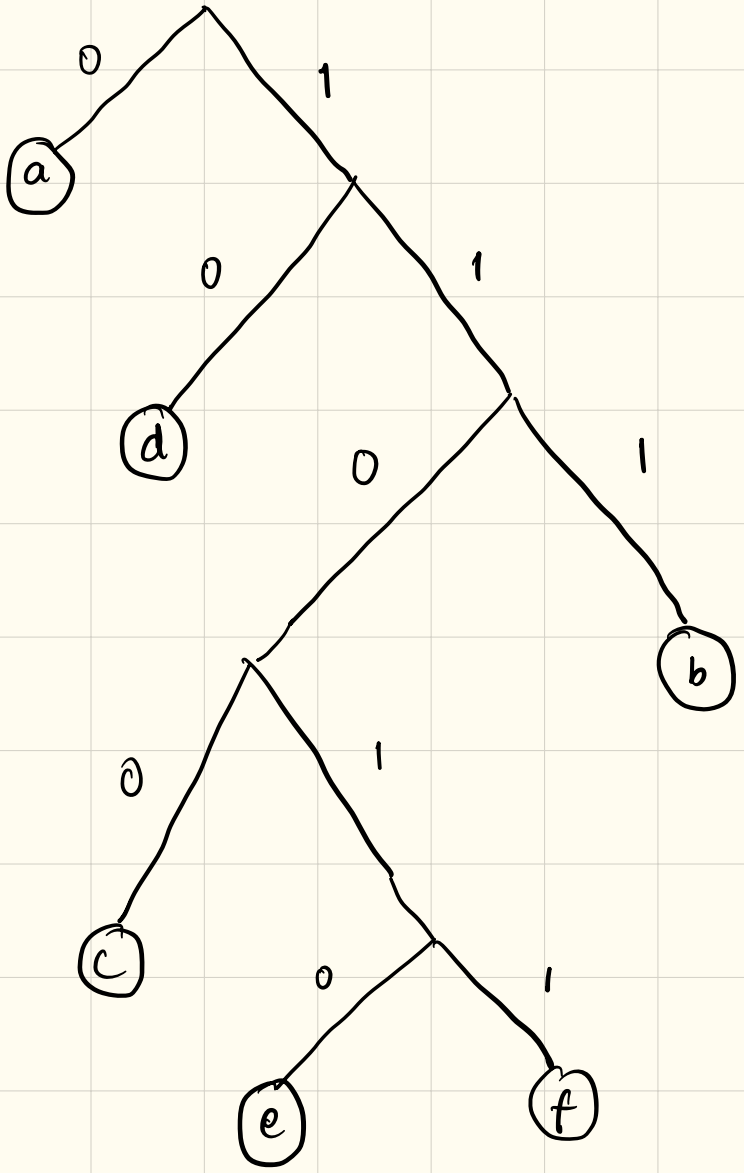
Bijection

Prefix-free encoding ↔ edge-labelled binary tree.

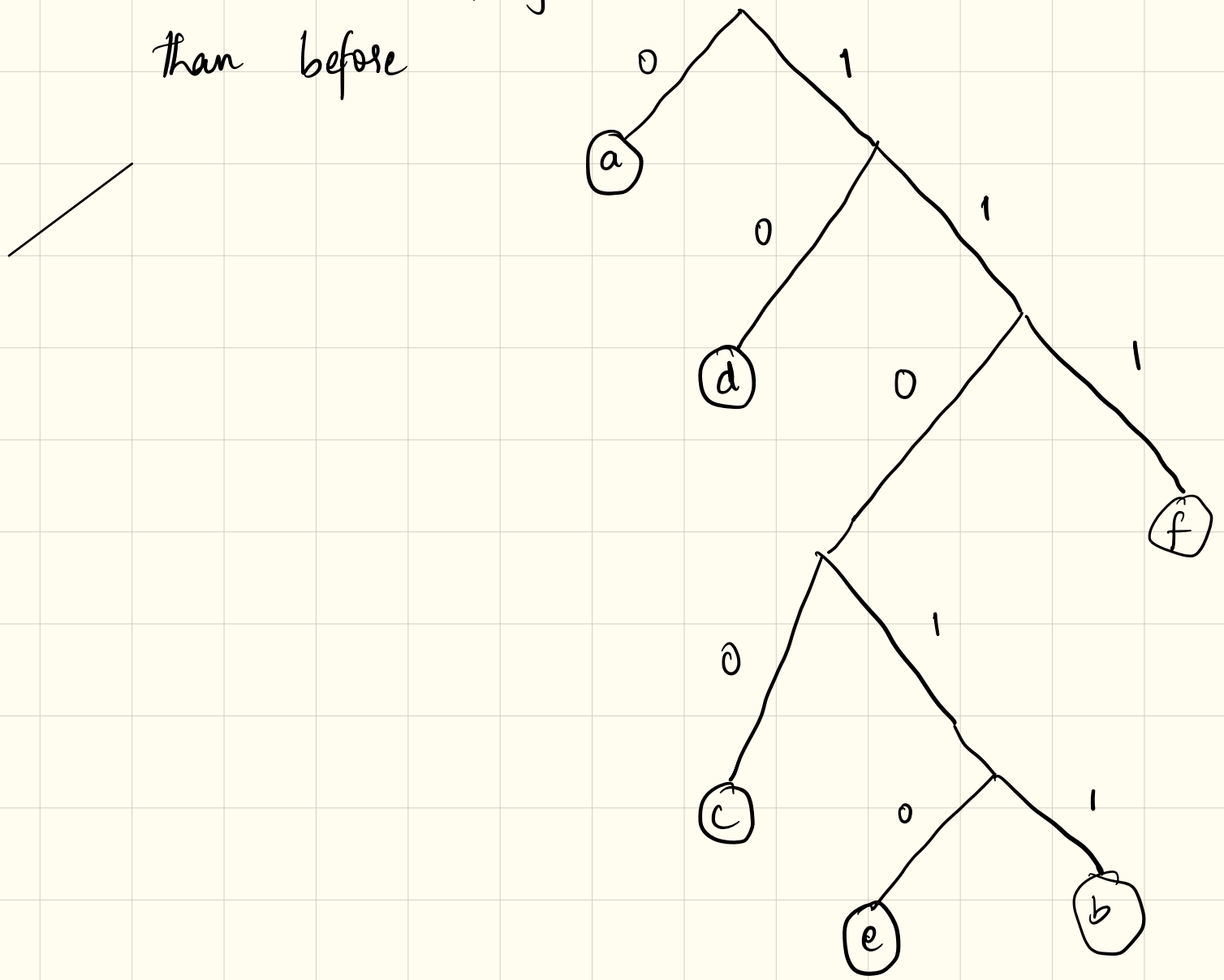


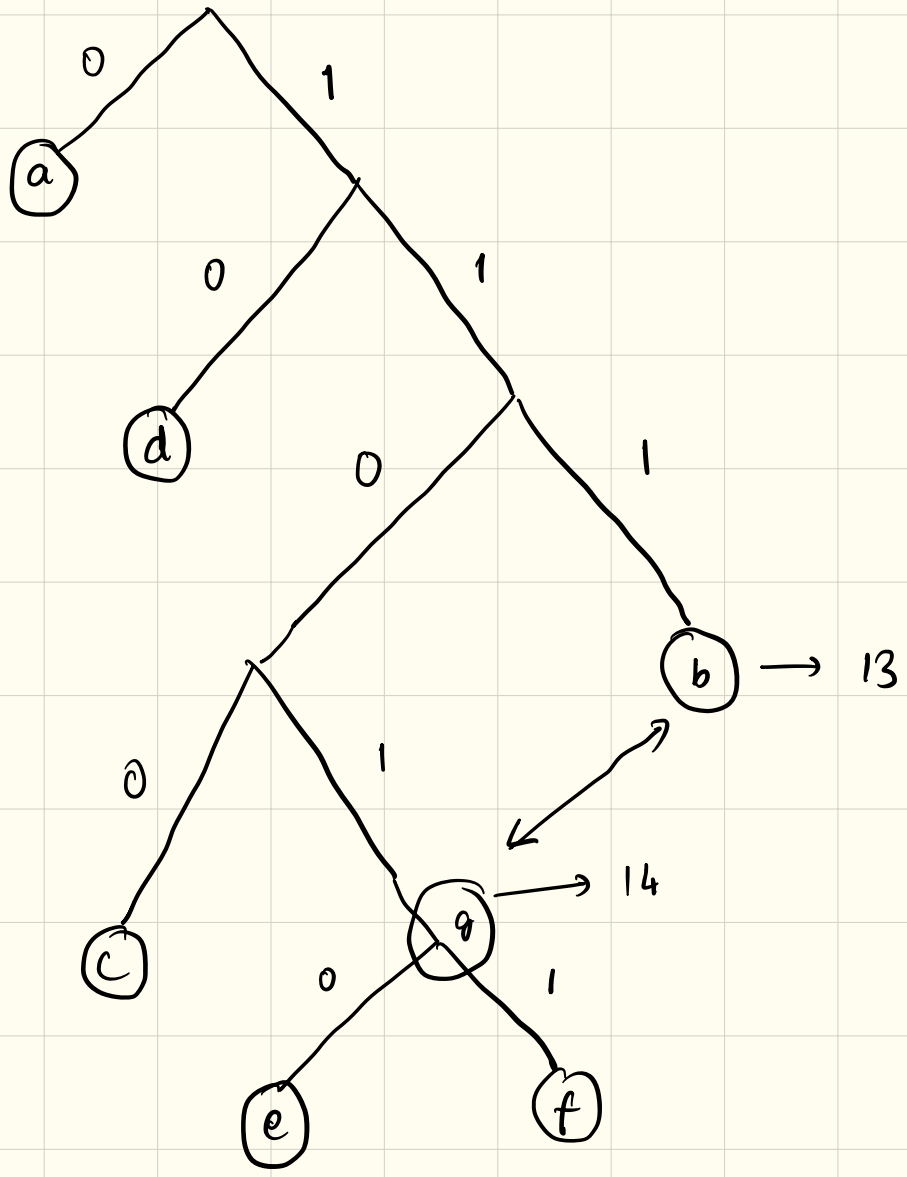
$$\text{bits}(T) = \sum_{c \in \Sigma^*} \text{freq}(c) \times \text{depth}(c)$$

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



Worse cost (higher)
than before

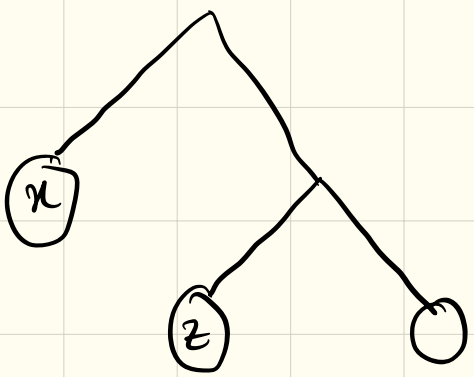



0
1
a
0
1
d
0
1
b
13
0
1
14
g
c
0
1
e
f

How to know if a tree is optimal?

Observation 1: (e) and (f) both will be leaves and siblings

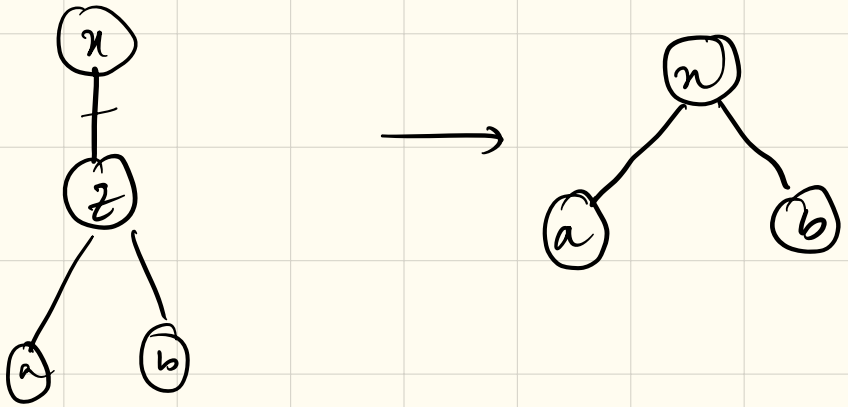
∃ an optimal tree where two least frequency characters are bottom-most leaves and siblings - (x, y)



freq (z) ≥ freq (x)

swapping x and z will clearly reduce cost.

* Every optimal tree will always have two children for internal nodes



* ∴ Optimal tree is always a binary tree.

06 Mar 2025 - Huffman Codes

Input :

Output :

a : 45	f : 5	Encoding each char. by 3 bits :
b : 13		300 bits
c : 12		
d : 16		
e : 9		

Prefix-free codes $\Rightarrow$ Unique decoding

Not (Codeword$_1$ is a prefix of Codeword$_2$) $\forall$ codewords,

Prefix codes $\longleftrightarrow$ full binary trees

↓

every internal node has 2 children

f : 5

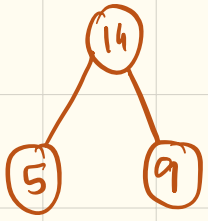
e : 9

c : 12

b : 13

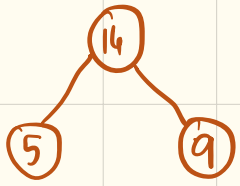
d : 16

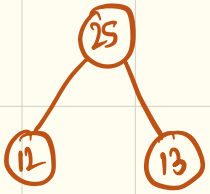
a : 45



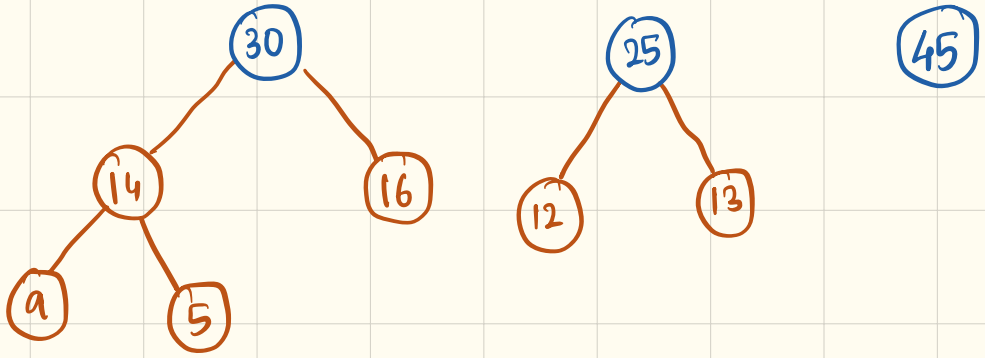
Step 1

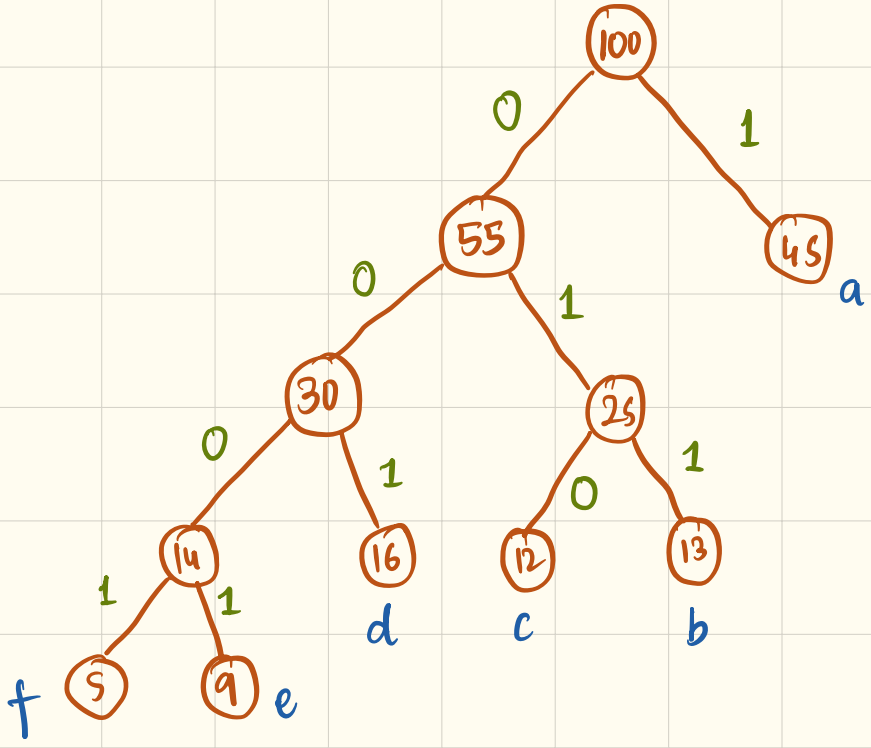
14 12 13 16 45





14 25 16 45





a : 1

b : 011

c : 010

d : 001

e : 0001

f : 0000

215 bits

* Min - heap

Running time : $O(n \log n)$

n = # of characters to encode

$2n$ Extract-min operation + n insert

using min - heaps

a : 1

b : 1

c : 2

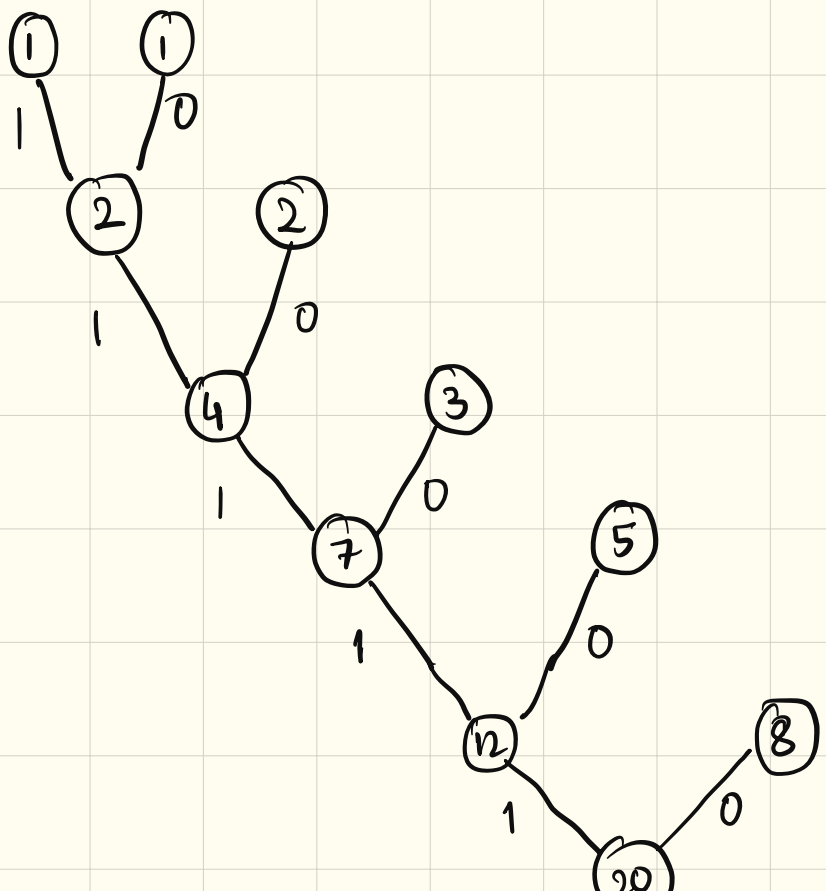
d : 3

e : 5

f : 8

g : 13

h : 21



h : 0

g : 10

f : 110

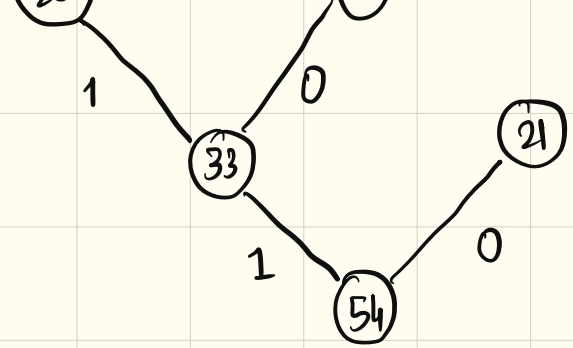
e : 1110

d : 11110

c : 111110

b : 1111110

a : 1111111



Cost : 132 bits

No Huffmann code : $2^8 = 256$ bits

Savings $\approx$ 48 %

Q1: We partition the set $\{1, \ldots, 100\}$ into k sets so that if a divides b, then a, b are in different sets. Min k

Q2: Partition into k decreasing subsequences. Min k.

1, 2, 10, 4, 11, 3, 7, 5, 6, 2

L → R

dec

1 divides all others $\{1\}$ $\{2, 3, \ldots\}$

2 divides all even no. $\{1\}$ $\{2, 4 \ldots\}$ $\{3, 5, 7, \ldots\}$

1 2 3 4

1 | 2 3 | 4 5 6 7 | 8 9 10 11 12 13 14

15 | 16 17 18 19 20 21 22 ... 31 | 32

$\underline{1} + \underline{2} + \underline{4} + \underline{8} + \underline{16} + \underline{32} + \underline{\sim}$

1 2 10 4 11 3 7 5 6 2

1

2 2

10 4 3

11 7 5

6

---

② {1}, {2}, {10, 4, 3, 2}, {11, 7, 5}, {6}

$\{ a_{i_1} > a_{\_} > a_{\_} > \cdots a_{i_2} \}$ $\quad i_1 < i_2$

$\{ a_{\_} > a_{\_} > \cdots \quad a_{i_2} \}$ $\quad < \cdots < i_k$

$\{ a_{\_} > a_{\_} > \cdots \quad a_{i_k} \}$

⋮

Claim: $a_{i_2} \leq a_{i_2} \leq \cdots \leq a_{i_k}$

Correctness ??

Running time

$\{ \cdots \quad b_1 \}$

$\{ \quad\quad b_2 \}$

⋮

$\{ \quad\quad b_\ell \}$

least j :

$b_j > a_i$

Binary search

$\{1\}$ $\{2, 3, 5, 7, \ldots\}$ primes

$\{4, 6, 9, 10\}$ product of 2 primes

$\vdots$

$\{1\}$ $\{2, 3\}$ $\{4, 5, 6, 7\}$