3335 Total characters in String After Transformations - 1

Input → String s
Integer t
↓
Number of transformation

Output → length of string after 't' transformations

Transformation Rules

→ Replace 'a' to 'y' with 'b' to 'z'
→ Replace 'z' with 'ab'

Example ①

⇒ S = abcyy , t = 2

output ⇒ 7

t=1 → bcdzz
t=2 → cdeabab → 7

Example (2)

$\Rightarrow$ S = azbk t = 1

O/p $\Rightarrow$ 5

Solution Approach

S = abcyy , t = 2

Various approach which are valid but are either

→ complex or

→ in-efficient

*) Brute-force → in-efficient

→ run a loop t time
   ↳ for t transformation

→ In each loop, run a loop over the string and replace the characters

→ Return length of final String

S = abcyy , t = 2

*) Count and update → complex slow

→ Intially count frequency of each character

[ ] → array of length 26

each index is Representation each character in lower case English letters

→ [1, 1, 1, ---- 2, 0]
↓ ↓ ↓ ↓ ↓
0 1 2 24 25
a b c y z

→ Run a loop t times

→ create a new frequency array of 26 length

→ Run a loop 25 times 'a' → 'y'

→ add the frequency in index 1 to 25

b to z

→ add frequency of z in indexes 0, 1

a b

→ frequency => new frequency

→ Return sum(frequency)

data[1] += z

↓ b

↓ add z to 'b'

⇒ [0, 1, 1, 1, ----- 0], z=2

0 1 2 3 ... 24

a b c d ... y

[2, 0, 1, 1, 1, ------ 0, 0]

0 1 2 3 4 ... 24 25

a b c d e ... y z

↓

+z

→ +2

[2, 2, 1, 1, 1 ------- 0, 0]

0 1 2 3 4 ... 24 25

a b c d e ... y z

o/p ⇒ sum

⇒ 2+2+1+1+1+0+0··· 0

⇒ (7) ✓

S ⇒ cde abab