Date: 13th January 2022

1. k Anagrams
2. Group Anagrams
3. Group Shifted Strings
4. word pattern
5. Isomorphic Strings

1. K Anagrams

k = 3

String s1 → a b a b a d d c c c

String s2 → b c b e c a b a c d

Suppose we are making s2 as s1

frequency map s1

a → 3
b → 2
d → 2
c → 2

frequency map of s2:

b → 2
c → 2
e → 1
a → 2
d → 1

s1 → ~~a3~~ ~~b2~~ ~~c2~~ ~~d2~~ ~~e0~~

s2 → ~~a2~~ ~~b3~~ ~~c3~~ ~~d1~~ ~~e1~~

+1 | -1 | 0 | +1 | -1

2 character change → s2 as s1.

& change req.

k=3

0, 1, 2, 3

Maximum change allowed = k

k = 1 → False

k = 3 → True

k = 2 → True

k=1

String s1 → a b a b a d d c c c]

String s2 → b c b e c a b a c d]

① if(length1 != length2)
↳ return False;

② Freq. map from s1

a → 2 2 1
b → 2 1 0
d → 2 1
c → 2 2 1 0

Count +ve chars

1 + 1 = 2

[Count <= k] true

2. Group Anagrams

Freq: map

Avg: O(1)

abcc → a1 b1 c2

acbc → a1 b1 c2

badc → a1 b1 c1 d1

abcd → a1 b1 c1 d1

dabb → a1 b2 d1

babd → a1 b2 d1

dbca → a1 b1 c1 d1

cabc → a1 b1 c2

abaa → a3 b1

key → value

a1 b1 c2 → L1 → [abcc, acbc, cabc],

a1 b1 c1 d1 → L2 → [badc, abcd, dbca],

a1 b2 d1 → L3 → [dabb, babd],

a3 b1 → L4 → [abaa]

no. of values

res = [L1, L2, L3, L4]

ArrayList < ArrayList < String > >

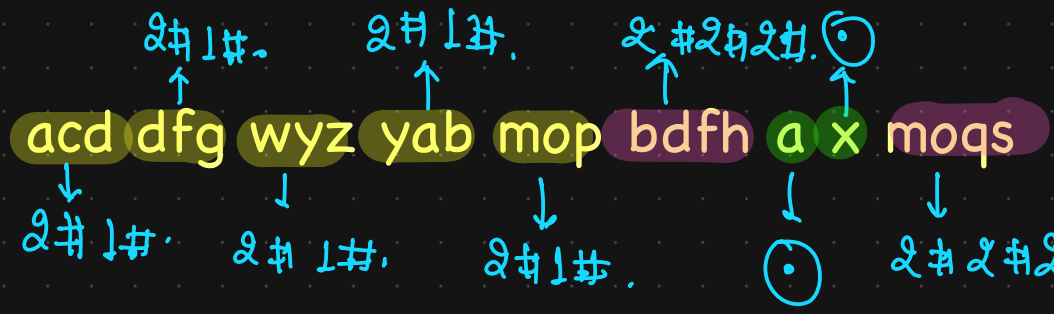
result

key → HashMap < Character, Integer >

freq. map

value → ArrayList < String >

3. Group Shifted Strings



y a b.

a-y = (-24) -ve.

+26 ↳ add 26 to make it positive

2

(Because of cyclic nature)

ab → (1)# bq

b - a = (1) # 1

1 - 2 = (-1) + 26 = (25) #

c1 c2 c3 c4 ... cn

C2-C1 # C3-C2 # C4-C3 # ... Cn-C(n-1) .

a → 1
b → 2
c → 3
d → 4
e → 5
f → 6
g → 7
h → 8
i → 9
j → 10
k → 11
l → 12
m → 13
n → 14
o → 15
p → 16
q → 17
r → 18
s → 19
t → 20
u → 21
v → 22
w → 23
x → 24
y → 25
z → 26

String code generate

key. → ALStrings

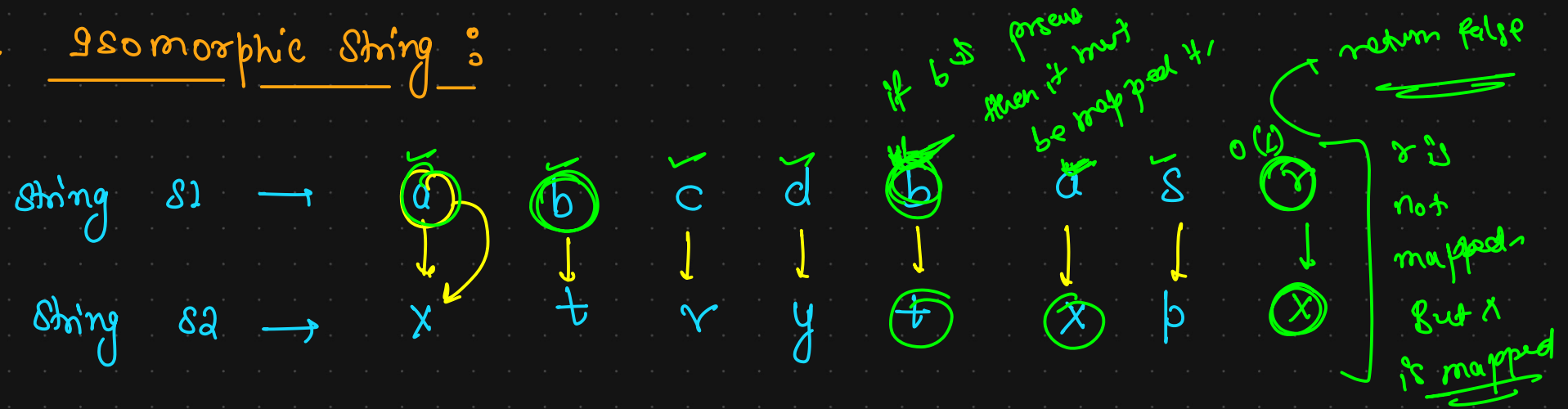
value.

a c f g m o q t

a → p

2 # 3 # 1 # 6 # 2 # 2 # 3 # .

4. Isomorphic String :



one to one mapping] → if present → true
→ Absent → false

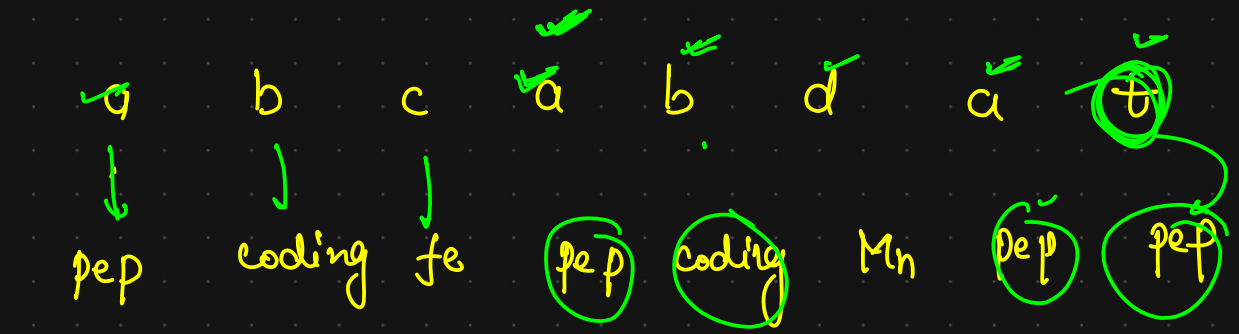
{ a → x
b → t
c → r
d → y

{ s → p

x is mapped
t is mapped
r is mapped
y is mapper.

p is mapper.

5. Word pattern :



a → pep
b → coding
c → fe
d → Mn

[map.containsValue(String)]

↳ O(n) to avoid time

we can use HashSet

↳ O(1)