

1st Jan 2022 (Saturday)

Tues, Thursday, Sat, Sun

2 batches \rightarrow PP10 + PP11

\hookrightarrow PP12

4c/weeks \rightarrow 3 hours

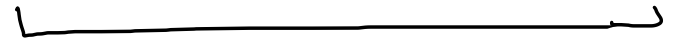
28 classes



6-7 weeks

(2 months)

8 6 3 4 6
A&S, SS, Tries, snQ, Bits,



11-2 \rightarrow 12

5-8 \rightarrow 7

DP

timings :

Tues, Thursday \rightarrow 5 to 8

Sat, Sun \rightarrow 11-2 or 5-8

K Anagrams

$k = 3$

s_1 : a b c c d a

s_2 : a a b j d j

$s_1 \rightarrow (c \rightarrow j) \times 2$

$s_1 \rightarrow$ a a b ~~c~~ ~~c~~ d

$s_2 \rightarrow$ a a b j j d

S1: a b a b c f a d k

S2: a b c c b f d m c

req. ops ≤ 1

S1: a a a b b c d f k

S2: a b b c c c d f m

a3 b2 c1 d1 f1 k1 m0

a1 b2 c3 d1 f1 k0 m1

Δ : 2 0 -2 0 0 1 -1

req. ops = 3

$\sum \Delta = 0$

$$\Delta' s = (a_1 - a_2) + (b_1 - b_2) + \dots + (z_1 - z_2)$$

$$= (a_1 + b_1 + c_1 + \dots + z_1) - (a_2 + b_2 + c_2 + \dots + z_2)$$

$$= \text{len}(s_1) - \text{len}(s_2)$$

$$= 0$$

S1 : a b a b c j a d k

a3 ... c1

S2 : a b c c b j d m c

a1 ... c3
2 -2

```
for(char ch : str1.toCharArray()) {
    int nf = map.getOrDefault(ch,0) + 1;
    map.put(ch,nf);
}

for(char ch : str2.toCharArray()) {
    int nf = map.getOrDefault(ch,0) - 1;
    map.put(ch,nf);
}

int ro = 0;
for(char key : map.keySet()) {
    int val = map.get(key);

    if(val > 0) {
        ro += val;
    }
}

return ro <= k;
```

S1 :

a - ~~3~~ 2

b - ~~2~~ 0

c - ~~1~~ ~~0~~ ~~-1~~ -2

d - ~~1~~ 0

j - ~~1~~ 0

k - 1

m - (-1)

ro = 2 + 1

49. Group Anagrams

Input: strs = ["eat", "tea", "tan", "ate", "nat", "bat"]

Output: [["bat"], ["nat", "tan"], ["ate", "eat", "tea"]]

[[tea, eat, ate] , [tan, nat] , [bat]]

 a-1
 e-1
 t-1

 a-1
 n-1
 t-1

 a-1
 b-1
 t-1

```
Input: strs = ["eat","tea","tan","ate","nat","bat"]
```

```
Output: [["bat"],["nat","tan"],["ate","eat","tea"]]
```

HM < map, AL < string > map;

hashmap {

equals,

hashCode

}

e-1
a-1
t-1 ↔ eat, tea, ate

t-1
a-1
n-1 → tan, nat

b-1
a-1
t-1 → bat

eg:

a b a b

c d o c

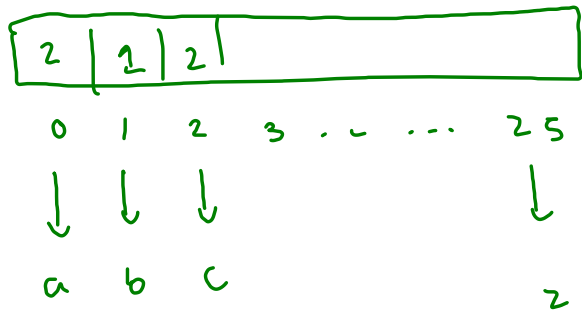
b a a b

a a b b

o d c c

get key

a2 b1 c2



get key (String str)?

[n → freq arr

[26 → encoded str

3

Group Shifted Strings

kmo, acj, nps, xtv, cde, lng

[kmo, xtv], [acj, nps, lng], [cde]

a	b	c	d	e
f	g	h	i	j
k	l	m	n	o
p	q	r	s	t
u	v	w	x	y
z				

a	b	c	d	e
f	g	h	i	j
k	l	m	n	o
p	q	r	s	t
u	v	w	x	y
z				

same shifted seq

$$\begin{array}{r}
 \begin{array}{ccc}
 \overset{c-a}{\text{a}} & & \overset{f-c}{\text{f}} \\
 & \text{c} &
 \end{array} \\
 \Delta : \quad \begin{array}{ccc} 1 & 1 & 1 \end{array} \\
 \hline
 \begin{array}{ccc}
 \overset{a+\Delta}{\text{b}} & \overset{c+\Delta}{\text{d}} & \overset{f+\Delta}{\text{g}} \\
 \text{c-a} & & \text{f-c}
 \end{array} \\
 \Delta : \quad \begin{array}{ccc} 3 & 3 & 3 \end{array} \\
 \hline
 \begin{array}{ccc}
 \overset{a+\Delta'}{\text{e}} & \overset{c+\Delta'}{\text{g}} & \overset{f+\Delta'}{\text{j}} \\
 \text{c-a} & & \text{f-c}
 \end{array}
 \end{array}$$

acj, bdg, egj

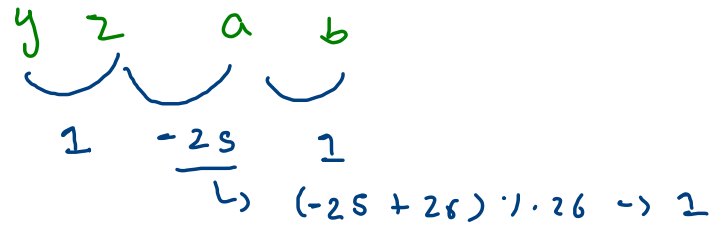
→ $\Delta' = 4$

kmo, acf, nps, stv, cdef, lng, yzab

a	b	c	d	e
f	g	h	i	j
k	l	m	n	o
p	q	r	s	t
u	v	w	x	y
z				

forward \rightarrow -ve

backward \rightarrow +ve



2 # 3 \rightarrow acf, nps, lng

2 # 2 \rightarrow stv, kmo

1 # 1 # 1 \rightarrow cdef, yzab