Group Anagrams

arr[] = { $\overset{\textcircled{1}}{\text{"cat"}}$, $\overset{\textcircled{2}}{\text{"dog"}}$, $\overset{\textcircled{1}}{\text{"tac"}}$, $\overset{\textcircled{2}}{\text{"god"}}$, $\overset{\textcircled{1}}{\text{"act"}}$ }

O/P "cat tac act dog god"

S_1 and S_2 are anagrams?

① Sorting TC: O(NlogN) SC: O(1)

② freq map of char TC: O(N)

SC: O(26) ⇝ O(1) }

arr[] = { "cat", "dog", "tac", "god", "act" }

Bruteforce : TC : $O(M^2 \times N)$

M → size of array

N → size the longest word

---

arr[] = { "cat", "dog", "tac", "god", "act" }

TC : $O(M \times N)$

freq Map (cat)

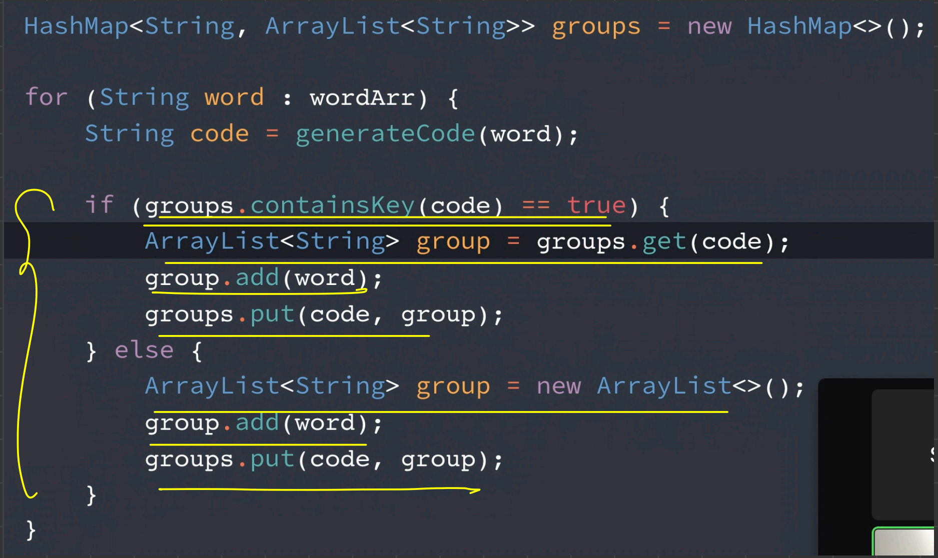
a →	1
c →	1
t →	1

→ "a1c1t1"

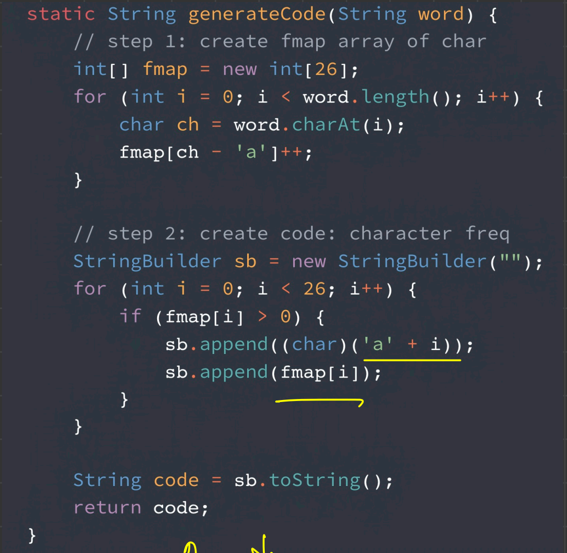
freq Map (tac)

a →	1
c →	1
t →	1

code	grp
a1c1t1	cat, tac, act-
d1g1o1	dog, god



```
HashMap<String, ArrayList<String>> groups = new HashMap<>();

for (String word : wordArr) {
    String code = generateCode(word);

    if (groups.containsKey(code) == true) {
        ArrayList<String> group = groups.get(code);
        group.add(word);
        groups.put(code, group);
    } else {
        ArrayList<String> group = new ArrayList<>();
        group.add(word);
        groups.put(code, group);
    }
}
```

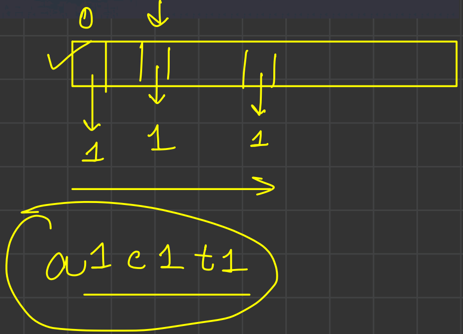


```
static String generateCode(String word) {
    // step 1: create fmap array of char
    int[] fmap = new int[26];
    for (int i = 0; i < word.length(); i++) {
        char ch = word.charAt(i);
        fmap[ch - 'a']++;
    }

    // step 2: create code: character freq
    StringBuilder sb = new StringBuilder("");
    for (int i = 0; i < 26; i++) {
        if (fmap[i] > 0) {
            sb.append((char)('a' + i));
            sb.append(fmap[i]);
        }
    }

    String code = sb.toString();
    return code;
}
```

arr[] = { "cat", "dog", "tac", "god", "act"}

d1g1o1

String	AL<String>
a1c1t1	{cat, tac, act}
d1g1o1	{dog, god}



Minimum Window Substring

Str1 d b a e c b b a b d c a a f b d d c a b g b a

Str2 a b b c d c

Brute force

TC : $O(N^3)$

SC : $O(N+M)$

inc

Str1 d b a e c b b a b d c a a f b d d c a b g b a

exc

mct = ~~0~~ ~~1~~ ~~2~~ ~~3~~ ~~4~~ 5

Str2 abbcdc

dmct = 6

freqMap

a	1
b	2
c	2
d	1

freqMap

d	1
b	~~1~~ ~~2~~ 3
a	1
c	1

pans