

Group Anagram

03 June 2025

08:25

We have list of strings, we need to store the group of strings together who possesses the anagrams.

eg list of strings = ['act', 'pots', 'tops',
'cat', 'stop', 'hat']

Output $\begin{bmatrix} ['act', 'cat'] & ['pots', 'tops', 'stop'] \\ & ['hat'] \end{bmatrix}$

We will count the characters in each string and store it in hashmap

eg. act & cut \rightarrow 3 characters

{ 'act': 3
cat: 3 }

Count =

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Diagram illustrating the counting process for the array. The array is divided into segments of 5 elements each, with indices 0 to 25. The count is 0 for all indices, indicating no elements are present in the array.

We have string = 'act'

Step 1: we loop through 'act'

↳ This will store
in default dict
as it is

When it sees 'a' $\rightarrow 0 \rightarrow 1$

see 'c' → 0 → 1
see 't' → 0 → 1 } similarly
for cat
also

Step 2: After storing the 26 values for word 'cat', it will go for the next word 'pots'. But before going to next word ~~it~~ count ~~is~~ will reset to zero again.

Then it follows same steps
for all words & save it default dict

Step 3: Lastly, return values of defaultdict in output.