

Group anagrams - hash table

class Solution:

```
def group_anagrams(self, strs: list[str]) -> list[list[str]]
```

```
    result = defaultdict(list)
```

```
    for s in strs:
```

```
        count = [0] * 26
```

```
        for char in s:
```

```
            count[ord(char) - ord("a")] += 1
```

```
        result[tuple(count)].append(s)
```

```
    return list(result.values())
```

```
print(Solution().group_anagrams(strs=["eat", "tea", "tan", "eat", "tea", "tan"]))
```

Step 1:

self -> solution instance

strs -> ["eat", "tea", "tan", "eat", "tea", "tan"]

Step 2:

result -> defaultdict <class 'list'>

Step 3: for s in strs
s → "act"

Step 4: count = [0] * 26
count = [0, 0, 0, 0, 0, 0, 0, 0, 0, ..., 0]

Step 5: for char in s:
char → "a"

(WFO: ord() = unicode of character → ord("a") = 97)

Step 6: count[ord("char") - ord("a")] += 1

count[ord("a") - ord("a")] += 1

count → [1, 0, 0, 0, 0, ..., 0] ^{26 times}

Step 7: for char in s:
char → "c"

Step 8: count[ord("char") - ord("a")] += 1

count[ord("c") - ord("a")] += 1

count[99 - 97] += 1

count → [1, 0, 1, 0, 0, 0, 0, 0, ..., 0]

Step 9 : for char in s:
char → "+"

Step 10: $\text{count}[\text{ord}(\text{"chae"}) - \text{ord}(\text{"a"})] += 1$

$$\text{count}[\text{ord}(t) - \text{ord}(a)] + 1$$
$$\text{count}[116 - 97] += 1$$

count[19] += 1

Count $\rightarrow [1, 0, 1, 0, 0, 0, 0, 0, 0, 0, \dots, 1, 0, 0, \dots]$

Step 11: `result [Tuple (count)].append(s)`

result $\rightarrow \{(1, 0, 1, 0, 0, 0, \dots, 1, 0, 00) : [act^1]\}$

Step 12: for s in s tes:
 $s \rightarrow \text{"pets"}$

$S \rightarrow \text{pats}$

Step 13: $\text{count} = [0]$ 26

Count $\rightarrow [0, 0, 0, 0, \dots, 0]$

Step 14: for char in s:

$$\text{char} \xrightarrow{0} \mathbb{F}$$

Step 15: $\text{count}[\text{ord}(p') - \text{ord}(a')] += 1$

$$\text{ord}("p") = 112$$

$$\text{count}[15] += 1$$

$$\text{count} \rightarrow [0, 0, 0, 0, \dots, \overset{15}{1}, \overset{16}{0}, \overset{17}{0}, \dots, 0]$$

Step 16: for char in s
char \rightarrow "o"

$$\text{Step 17: } \text{count}[\text{ord}("o") - \text{ord}("a")] += 1$$

$$\text{ord}("o") = 111$$

$$\text{count}[14] += 1$$

$$\text{count} \rightarrow [0, 0, 0, 0, \dots, \overset{14}{1}, \overset{15}{1}, 0, 0, 0]$$

Step 18: for char in s
char \rightarrow "t"

$$\text{Step 19: } \text{count}[\text{ord}(\text{char}) - \text{ord}("a")] += 1$$

$$\text{count}[19] += 1$$

$$\text{count} \rightarrow [0, 0, 0, 0, \dots, \overset{14,15}{1, 1}, \overset{16}{0}, \overset{17}{0}, \overset{18}{0}, \overset{19}{1}, 0, 0]$$

Step 20: for char in s
char \rightarrow "s"

Step 21: $\text{count}[\text{ord}(\text{char}) - \text{ord}('a')] += 1$

$\text{ord}('s') = 115$

$\text{count} \rightarrow [0, 0, 0, 0, \dots, 1, 1, 0, 0, 1, 1, 0, 0, 0]$
 (indices 14, 15, 18, 19, 20)

Step 22: $\text{result}[\text{tuple}(\text{count})].\text{append}(s)$

$\text{result} \rightarrow \{ (1, 0, 1, 0, 0, 0, \dots, 1, 0, 0) : ["act"], (0, 0, 0, 0, \dots, 1, 1, \dots, 0, 0, 0, 1, 1, \dots) : ["pets"] \}$

• this will be the same for the following words: "tops", "cat", "step" where result will be:

$\text{result} \rightarrow \{ (1, 0, 1, 0, \dots, 0, 1, 0, 0) : ["act", "cat"]$

$(0, 0, 0, \dots, 1, 1, \dots, 0, 0, 1, 1, \dots, 0, 0) : ["pets", "tops", "step"] \}$

Step x: for s in strs: $\text{count} = [0] \times 26$

$s \rightarrow "hat"$

$\text{count} \rightarrow [0, 0, 0, 0, 0, 0, 0, \dots, 0]$

Step y: for char in s:

$\text{count}[\text{ord}(\text{char}) - \text{ord}('a')] += 1$

$\text{char} \rightarrow "h"$

$\text{ord}('h') = 104$

$\text{count}[104 - 97] += 1$

count [0, 0, 0, ..., ⁷1, ..., 0, 0, ..., 0]
- for "t" and "a" characters :
count [1, 0, 0, ..., ⁷1, ..., 0, 0, ¹³1, 0, 0, ..., 0]

result will become \rightarrow result [tuple(count)] .append(s)
(1, 0, 1, 0, ..., 0, ..., 1, 0, 0) : ["act", "cat"],
(0, 0, 0, ..., ⁷1, ¹³1, ..., 0, 0, 1, 1, ..., 0, 0) : ["pots", "tops", "stop"],
(1, 0, 0, ..., 1, ..., 0, 0, 1, ..., 0) : ["hot"]

Step 2: list(result.values())

last result \rightarrow [["act", "cat"], ["pots", "tops", "stop"], ["hot"]]

