

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

### **Step 1:**

The word “eat” :

[]

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
1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

Sorting alphabetically is [aet]

Updating the hash table:

Key : 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0

Value : eat

### **Step 2:**

The word “tea” :

["eat"]

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
1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

Sorting alphabetically is [aet]

Which becomes ["eat", "tea"]

Updating the hash table:

Key : 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0

Value : tea

### **Step 3:**

The word “tan” :

[]

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
1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0

Sorting alphabetically is [ant]

Which becomes [“tea”]

Updating the hash table:

Key : 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0

Value : tea

### **Step 4:**

The word “ate” :

[]

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
1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

Sorting alphabetically is [aet]

Which becomes [“eat”, “tea”, “ate”]

Updating the hash table:

Key : 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0

Value : “ate”

### **Step 5:**

The word “nat” :

[]

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
1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0

Sorting alphabetically is [ant]

Which becomes ["tan", "nat"]

Updating the hash table:

Key : 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0

Value : nat

### **Step 6:**

The word “bat” :

[]

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
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

Sorting alphabetically is [abt]

Which becomes ["bat"]

Updating the hash table:

Key : 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0

Value : bat

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

Visual Studio Code interface showing a Python file named `anagrams.py` in the `WEEK3 ASSY` folder. The code defines a `Solution` class with a `groupAnagrams` method that groups words by their sorted characters.

```
1 class Solution(object):
2     def groupAnagrams(self, strs) -> list:
3         anagrams: dict = {}
4
5         for word in strs:
6             key: LiteralString = ''.join(sorted(word))
7             if key in anagrams:
8                 anagrams[key].append(word)
9             else:
10                anagrams[key] = [word]
11
12        result: list = list(anagrams.values())
13        return result
14
15
16 #test case
17 #Input: strs = ["eat", "tea", "tan", "ate", "nat", "bat"]
18 strs: list[str] = ["eat", "tea", "tan", "ate", "nat", "bat"]
19 print(Solution().groupAnagrams(strs))
20
21
```

The terminal output shows the execution of the script:

```
PS C:\Users\melan\Desktop\Algorithms\week3 assy> cd "c:/Users/melan/Desktop/Algorithms/week3 assy"
PS C:\Users\melan\Desktop\Algorithms\week3 assy> & C:/Users/melan/AppData/Local/Microsoft/WindowsApps/python3.10.exe "c:/Users/melan/Desktop/Algorithms/week3 assy/anagrams.py"
[['eat', 'tea', 'ate'], ['tan', 'nat'], ['bat']]
PS C:\Users\melan\Desktop\Algorithms\week3 assy>
```

The status bar at the bottom indicates the file is at line 17, column 58, using UTF-8 encoding. The system tray shows the date and time as 4:32 PM on 3/29/2023.

## CODE

```
class Solution(object):
    def groupAnagrams(self, strs):
        anagrams = {}

        for word in strs:
            key = ''.join(sorted(word))
            if key in anagrams:
                anagrams[key].append(word)
            else:
                anagrams[key] = [word]

        result = list(anagrams.values())
        return result

#test case
#Input: strs = ["eat", "tea", "tan", "ate", "nat", "bat"]
strs = ["eat", "tea", "tan", "ate", "nat", "bat"]
print(Solution().groupAnagrams(strs))
```