Least Frequent Character in String

15 September 2025 09:46 PM

The task is to find the least frequent character in a string, we count how many times each character appears and pick the one with the lowest count.

Using collections.Counter

The most efficient way to do this is by using collections.Counter which counts character frequencies in one go and makes it easy to find the least frequent character.


```
from collections import Counter
s = "GeeksforGeeks"
freq = Counter(s)
res = min(freq, key=freq.get)
print(str(res))
```

Output

f

Explanation:

- **Counter(s)**: This creates a dictionary where each character in the string is a key, and the value is its frequency.
- **min(freq, key=freq.get)**: This finds the character with the lowest frequency.

Table of Content

Using dict

Using a dictionary, we count the frequency of each character in a single pass and directly find the least frequent character. This approach avoids re-scanning the string making it more efficient than naive methods.


```
s = "GeeksforGeeks"
d = {}  # Initialize an empty dictionary

for char in s:  # Loop through each character in `s`
    d[char] = d.get(char, 0) + 1
res = min(d, key=d.get)
print(str(res))
```

Output

f

Explanation:

- **d.get(char, 0):** This retrieves the current frequency of char, defaulting to 0 if the character is not found.
- **min(d, key=d.get):** This finds the character with the lowest frequency in the dictionary ***d***.

Using Sorting

Using sorting, we first count the frequency of each character in the string and then sort the characters by their frequency in ascending order. The least frequent character is extracted from the first element of the sorted list.


```
s = "GeeksforGeeks"
# Creating an empty dictionary
d = {}

# Counting the frequency of each character in the string
for char in s:
    d[char] = d.get(char, 0) + 1

# Sorting `d` items by frequency
sorted_freq = sorted(d.items(), key=lambda item: item[1])

# Extracting the least frequent character
res = sorted_freq[0][0]
print(res)
```

Output

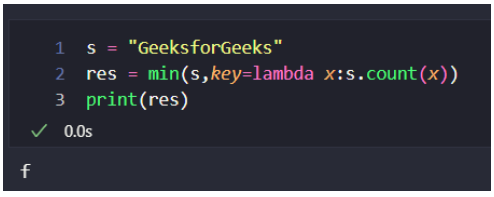
f

Explanation:

- **d[char] = d.get(char, 0) + 1**: This checks if the character already exists in the dictionary ***d*** . If it does it increments its count, otherwise it initializes the count to 1.
- **sorted(d.items(), key=lambda item: item[1])**: This sorts the items in the dictionary by the frequency in ascending order.
- **sorted_freq[0][0]**: After sorting, the least frequent character is the first element in the sorted list***sorted_freq***and its character is extracted using [0][0].

Using str.count()

count()function counts the occurrences of each character and min() is used to identify the character with the lowest frequency. This approach is straightforward but becomes inefficient for larger strings due to its quadratic time complexity.




```
s = "GeeksforGeeks"
res = min(s, key=lambda char:
          s.count(char))
print(str(res))
```

Output

f

Explanation:

- **min(s, key=...):** This finds the character with the minimum frequency in the string ***s*** .
- **lambda char: s.count(char):** For each character this lambda function calculates the frequency of that character in the string using s.count(char).

completed

From <https://www.geeksforgeeks.org/python/python-least-frequent-character-in-string/>