# Strings – Character Frequency

## 1. Introduction

**Character frequency** refers to counting **how many times each character appears** in a string.

It is a fundamental string operation and is widely used in **text analysis, pattern matching, and data processing**.

Understanding character frequency helps learners develop:

- Traversal logic
- Counting techniques
- Use of auxiliary storage

---

## 2. What is Character Frequency?

Given a string, character frequency tells **how often each character occurs**.

### Example:

```
String: "banana"
```

Character counts:

- b → 1
- a → 3
- n → 2

---

## 3. Why is Character Frequency Important?

Character frequency is important because:

- It helps analyze textual data
- Used in anagram checking

- Used in compression algorithms
- Required in many interview questions
- Forms the base for advanced string problems

---

## 4. Basic Idea Behind Character Frequency

The basic idea is:

- Traverse the string character by character
- Maintain a count for each character
- Update the count whenever the character appears

This requires an **auxiliary data structure** to store counts.

---

## 5. Logic for Character Frequency (Plain English)

1. Start from the first character of the string
2. Read one character at a time
3. If the character is seen for the first time, initialize its count
4. If the character already exists, increase its count
5. Continue until the end of the string
6. Display each character with its frequency

---

## 6. Visualization Example

Given string:

```
"APPLE"
```

Traversal process:

- Read A → count = 1
- Read P → count = 1

- Read P → count = 2
- Read L → count = 1
- Read E → count = 1

Final result:

```
A:1  P:2  L:1  E:1
```

---

# 7. Methods to Find Character Frequency

## 1️⃣ Using Array (ASCII Based)

- Create an array of size 26 or 256
- Map characters to indices
- Fast and memory-efficient

## 2️⃣ Using Hash Map / Dictionary

- Store characters as keys
- Store counts as values
- Flexible and easy to implement

---

# 8. Time and Space Complexity

| Aspect | Complexity |
|---|---|
| Time Complexity | O(n) |
| Space Complexity | O(1) or O(k) |

Where n is string length and k is number of unique characters.

---

# 9. Special Cases

- **Empty string:** No characters to count

- **Case sensitivity:** A and a are treated differently
- **Spaces and symbols:** Can be included or ignored based on requirement
- **Single character string:** Frequency is 1

---

## 10. Real-World Applications

- Word frequency analysis
- Password strength checking
- Text compression
- Data analytics
- Natural language processing (basic)

---

## 11. Common Problems Using Character Frequency

- Anagram check
- First non-repeating character
- Most frequent character
- Remove duplicate characters
- Count vowels and consonants

---

## 12. Summary

- Character frequency counts occurrences of each character
- Traversal is required
- Uses auxiliary storage
- Time complexity is O(n)
- Important for text-based problems

---