<https://github.com/doocs/leetcode/blob/main/solution/1100-1199/1119.Remove%20Vowels%20from%20a%20String/README_EN.md>

Given a string s, remove the vowels 'a', 'e', 'i', 'o', and 'u' from it, and return the new string.

**Example 1:**

**Input:** s = "leetcodeisacommunityforcoders"

**Output:** "ltcdscmmntyfrcdrs"

**Example 2:**

**Input:** s = "aeiou"

**Output:** ""

**Constraints:**

1 <= s.length <= 1000

s consists of only lowercase English letters.

**Attempt 1: 2025-7-5**

**Solution 1: String (10 min)**

**Refer to Deepseek**

**Problem Statement**

Given a string S, remove the vowels 'a', 'e', 'i', 'o', 'u' (both lowercase and uppercase) from it, and return the new string.

**Approach**

**Iterate through each character** in the string

**Check if the character is a vowel** (both lowercase and uppercase)

**Build a new string** containing only non-vowel characters

Return the filtered string

**Solution Code**

class Solution {

public String removeVowels(String S) {

StringBuilder result = new StringBuilder();

String vowels = "aeiouAEIOU";

for (char c : S.toCharArray()) {

if (vowels.indexOf(c) == -1) { // Not a vowel

result.append(c);

}

}

return result.toString();

}

}

**Alternative Solution (Using Regex)**

class Solution {

public String removeVowels(String S) {

return S.replaceAll("[aeiouAEIOU]", "");

}

}

**Explanation**

**StringBuilder Approach**:

Creates a StringBuilder to efficiently build the result string

Checks each character against a string containing all vowels

Only appends non-vowel characters to the result

Converts the StringBuilder to a string before returning

**Regex Approach**:

Uses String.replaceAll() with a regex pattern matching any vowel

More concise but slightly less efficient due to regex processing

**Complexity Analysis**

**Time Complexity**: O(n) where n is the length of the string

Both approaches require a single pass through the string

**Space Complexity**: O(n) for the output string

In the worst case (no vowels removed), we need space equal to input

Both solutions are efficient for this problem, with the StringBuilder approach being generally preferred for string manipulation tasks in Java.

**Refer to**

[L345.Reverse Vowels of a String (Ref.L344,L1119)](note://CA879E2192A94099996642D7560B6FB9)