1768. Merge Strings Alternately

Hint

Easy 2.8K 49

Companies

You are given two strings `word1` and `word2`. Merge the strings by adding letters in alternating order, starting with `word1`. If a string is longer than the other, append the additional letters onto the end of the merged string.

Return *the merged string*.

Example 1:

```
Input: word1 = "abc", word2 = "pqr"
Output: "apbqcr"
Explanation: The merged string will be merged as so:
word1:  a   b   c
word2:    p   q   r
merged: a p b q c r
```

Example 2:

```
Input: word1 = "ab", word2 = "pqrs"
Output: "apbqrs"
Explanation: Notice that as word2 is longer, "rs" is appended to the end.
word1:  a   b
word2:    p   q   r   s
merged: a p b q   r   s
```

Example 3:

```
Input: word1 = "abcd", word2 = "pq"
Output: "apbqcd"
Explanation: Notice that as word1 is longer, "cd" is appended to the end.
word1:  a   b   c   d
word2:    p   q
merged: a p b q c   d
```

Constraints:

- `1 <= word1.length, word2.length <= 100`
- `word1` and `word2` consist of lowercase English letters.

Accepted 363.3K | Submissions 455.6K | Acceptance Rate 79.7%

Approach 1: Using two pointers
one for word1
one for word2

```
string ans
i=0 , j=0
while( i<w1.size() && j<w2.size())
{   ans = ans + w1[i] + w2[j]
    i++ , j++
}
while( i<w1.size())   ans = ans + w1[i++]
while( j<w2.size())   ans = ans + w2[j++]
```

Instead of 3 loops we can just have one loop so that the code is smaller.

```
while( i<w1.size() || j<w2.size())
{
    if(i<w1.size())
        ans = ans + w1[i] , i++
    if(j<w2.size())
        ans = ans + w2[j] , j++
}
```

Instead of 2 pointers we can also do using 1 pointer.

```
for(i=0 ; i<max(w1.size,w2.size) ;i++)
{
    if(i<w1.size)
        ans = ans + w1[i]
    if(i<w2.size)
        ans = ans + w2[i]
}
```

$\Gamma(m) = O(w_1 + w_2)$