Number of Lines To Write String

We are to write the letters of a given string `S`, from left to right into lines. Each line has maximum width 100 units, and if writing a letter would cause the width of the line to exceed 100 units, it is written on the next line. We are given an array `widths`, an array where widths[0] is the width of 'a', widths[1] is the width of 'b', ..., and widths[25] is the width of 'z'.

Now answer two questions: how many lines have at least one character from `S`, and what is the width used by the last such line? Return your answer as an integer list of length 2.

Example :
Input:
```
widths = [10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10]
S = "abcdefghijklmnopqrstuvwxyz"
```
Output: [3, 60]
Explanation:
All letters have the same length of 10. To write all 26 letters,
we need two full lines and one line with 60 units.

Example :
Input:
```
widths = [4,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10]
S = "bbbcccdddaaa"
```
Output: [2, 4]
Explanation:
All letters except 'a' have the same length of 10, and
"bbbcccdddaa" will cover 9 * 10 + 2 * 4 = 98 units.
For the last 'a', it is written on the second line because
there is only 2 units left in the first line.
So the answer is 2 lines, plus 4 units in the second line.

Note:

- The length of `S` will be in the range [1, 1000].
- `S` will only contain lowercase letters.
- `widths` is an array of length `26`.
- `widths[i]` will be in the range of `[2, 10]`.

Solution 1

Very self-explaining codes.

Note from description:

- The length of S will be in the range `[1, 1000]`.
- S will only contain lowercase letters.
- widths is an array of length 26.

C++

```
vector<int> numberOfLines(vector<int>& widths, string S) {
    int res = 1, cur = 0;
    for (char c : S) {
        int width = widths[c - 'a'];
        res = cur + width > 100 ? res + 1 : res;
        cur = cur + width > 100 ? width : cur + width;
    }
    return {res, cur};
}
```

Java:

```
public int[] numberOfLines(int[] widths, String S) {
    int res = 1, cur = 0;
    for (char c : S.toCharArray()) {
        int width = widths[c - 'a'];
        res = cur + width > 100 ? res + 1 : res;
        cur = cur + width > 100 ? width : cur + width;
    }
    return new int[] {res, cur};
}
```

Python

```
def numberOfcurs(self, widths, S):
    res, cur = 1, 0
    for i in S:
        width = widths[ord(i) - ord('a')]
        res += 1 if cur + width > 100 else 0
        cur = width if cur + width > 100 else cur + width
    return [res, cur]
```

written by lee215 original link here

Solution 2

```ruby
def number_of_lines(widths, s)
  a = s.gsub(/./) { "a#{'.' * (widths[$&.ord - 97] - 2)}b" }.scan(/a.{,98}b/)
  [a.size, a[-1].size]
end
```

I replace for example a letter of width `6` by the string `a....b`. Then use regex `/a.{,98}b/` to create optimal lines.

written by StefanPochmann original link here

Solution 3

```
vector<int> numberOfLines(vector<int>& widths, string S) {
    int lines = 1; vector<int> res;
    int occupied_width = 0, len = S.length();

    for(int i = 0; i < len; i++) {
        if((occupied_width + widths[S[i] - 'a']) > 100) {
            lines++;
            occupied_width = 0;
        }
        occupied_width += widths[S[i] - 'a'];
    }

    res.push_back(lines);
    res.push_back(occupied_width);

    return res;
}
```

written by DDev original link here