763. Partition Labels

You are given a string `s`. We want to partition the string into as many parts as possible so that each letter appears in at most one part.

Return *a list of integers representing the size of these parts*.

Example 1:

```
Input: s = "ababcbacadefegdehijhklij"
Output: [9,7,8]
Explanation:
The partition is "ababcbaca", "defegde", "hijhklij".
This is a partition so that each letter appears in at most one part.
A partition like "ababcbacadefegde", "hijhklij" is incorrect, because it
splits s into less parts.
```

Example 2:

```
Input: s = "eccbbbbdec"
Output: [10]
```

Constraints:

- `1 <= s.length <= 500`
- `s` consists of lowercase English letters.

```
class Solution:
    def partitionLabels(self, s: str) -> List[int]:
        freq = {}
        for i,ele in enumerate(s):
            freq[ele] = i
        prev = -1
        maxIdx = 0
        ans = []

        for i in range(len(s)):
            maxIdx = max(freq[s[i]],maxIdx)
            if maxIdx == i:
                ans.append(maxIdx-prev)
                prev = maxIdx
        return ans
```