

763. Partition Labels

[Description](#)[Hints](#)[Submissions](#)[Discuss](#)[Solution](#)

- Difficulty:Medium
- Total Accepted:3.4K
- Total Submissions:5K
- Contributor:[awice](#)
- [Subscribe](#) to see which companies asked this question.

Related Topics

String Greedy

A string *S* of lowercase letters is given. We want to partition this string into as many parts as possible so that each letter appears in at most one part, and 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.

Note:

1. *S* will have length in range [1, 500].
2. *S* will consist of lowercase letters ('a' to 'z') only.

Seen this question in a real interview before?

```
class Solution {
```

```
public:
```

```
    vector<int> partitionLabels(string S)
```

```
{
```

```
    vector<int> res;
```

```
    vector<int> pos(26,0);
```

```
    for(int i=0;i<(int)S.size();i++)
```

```
    {
```

```
        pos[S[i]-'a'] = i;
```

```
}  
  
int end = -1;  
  
int start = -1;  
  
for(int i=0;i<(int)S.size();++i)  
{  
  
    if(start==-1) start=i;  
  
    end = max(end,pos[S[i]-'a']);  
  
    if(end==i)  
    {  
  
        res.push_back(end-start+1);  
  
        start=-1;  
  
    }  
  
}  
  
return res;  
  
}  
  
};
```