471. Encode String with Shortest Length

Add to List

Description (?tab=Description) | Submission (?tab=Submission) | Solutions (?tab=Solutions)

Total Accepted: **2628** Total Submissions: **6262** Difficulty: **Hard** Contributors: **Admin**

Notes

Given a **non-empty** string, encode the string such that its encoded length is the shortest.

The encoding rule is: `k[encoded_string]` , where the *encoded_string* inside the square brackets is being repeated exactly *k* times.

Note:

1. *k* will be a positive integer and encoded string will not be empty or have extra space.
2. You may assume that the input string contains only lowercase English letters. The string's length is at most 160.
3. If an encoding process does not make the string shorter, then do not encode it. If there are several solutions, return any of them is fine.

Example 1:

```
Input: "aaa"
Output: "aaa"
Explanation: There is no way to encode it such that it is shorter than the input string, so we do not encode it.
```

Example 2:

```
Input: "aaaaa"
Output: "5[a]"
Explanation: "5[a]" is shorter than "aaaaa" by 1 character.
```

Example 3:

```
Input: "aaaaaaaaaa"
Output: "10[a]"
Explanation: "a9[a]" or "9[a]a" are also valid solutions, both of them have the same length = 5, which is the same as "10[a]".
```

Example 4:

```
Input: "aabcaabcd"
Output: "2[aabc]d"
Explanation: "aabc" occurs twice, so one answer can be "2[aabc]d".
```

Example 5:

```
Input: "abbbabbbcabbbabbbc"
Output: "2[2[abbb]c]"
Explanation: "abbbabbbc" occurs twice, but "abbbabbbc" can also be encoded to "2[abbb]c", so one answer can be "2[2[abbb]c]".
```

Hide Company Tags | Google (/company/google/)

Hide Tags | Dynamic Programming (/tag/dynamic-programming/)

Hide Similar Problems | (M) Decode String (/problems/decode-string/)

Have you met this question in a real interview? Yes No

Discuss (https://discuss.leetcode.com/category/602) | Pick One (/problems/random-one-question/) | Editorial Solution

C++

```
class Solution {
public:

    // So Difficult
    // https://discuss.leetcode.com/topic/71541/easy-to-understand-c-o-n-3-solution


    // check this question Repeated Substring Pattern for collapse function
```

Send Feedback (mailto:admin@leetcode.com?subject=Feedback)

```
    // https://discuss.leetcode.com/topic/68206/easy-python-solution-with-explaination
    string collapse(string& s, int i, int j, vector<vector<string>>& dp) {
        string t = s.substr(i, j - i + 1);
        auto pos = (t + t).find(t, 1);
        if(pos < t.size())
            return to_string(t.size() / pos) + "[" + dp[i][i+pos-1] + "]";
        return t;
    }


    string encode(string s) {
        int N = s.length();
        vector<vector<string>> dp(N, vector<string>(N, ""));
        for(int len = 1; len <= s.length(); ++len) {
            for(int i = 0; i + len <= N; ++i) {
                int j = i + len - 1;
                dp[i][j] = s.substr(i, len);
                for(int k = i; k < j; ++k) {
                    if(dp[i][k].size() + dp[k+1][j].size() < dp[i][j].size()) {
                        dp[i][j] = dp[i][k] + dp[k+1][j];
                    }
                }
                string replace = collapse(s, i, j, dp);
                if(replace.size() < dp[i][j].size()) {
                    dp[i][j] = replace;
                }
            }
        }
        return dp[0][N-1];
    }
```

Notes

Custom Testcase

Contribute Testcase

Run Code | Submit Solution