

[6. Z 字形变换]: 【代码优化】

题目描述:

将一个给定字符串 `s` 根据给定的行数 `numRows`，以从上往下、从左到右进行 Z 字形排列。

比如输入字符串为 `"PAYPALISHIRING"` 行数为 `3` 时，排列如下：

```
P   A   H   N
A P L S I I G
Y   I   R
```

之后，你的输出需要从左往右逐行读取，产生出一个新的字符串，比如：`"PAHNAPLSIIGYIR"`。

请你实现这个将字符串进行指定行数变换的函数：

```
string convert(string s, int numRows);
```

示例 1:

输入: `s = "PAYPALISHIRING", numRows = 3`
输出: `"PAHNAPLSIIGYIR"`

示例 2:

输入: `s = "PAYPALISHIRING", numRows = 4`
输出: `"PINALSIGYAHRPI"`

解释:

```
P       I       N
A  L S   I G
Y A    H R
P       I
```

示例 3:

输入: `s = "A", numRows = 1`
输出: `"A"`

题目分析

1: 首先肯定是读懂题意

2: 根据给出的示例考虑各种特殊的情况 (分析的严谨性) --此处代码显得冗长

```
class Solution {
public:
    string convert(string s, int numRows) {
        //特殊情况处理
        if(numRows==1) return s;
        vector<string>res(numRows, "");
        string ans;
        //特殊情况处理
        if(numRows==2) {
            for(int i=0;i<s.size();i++){
                if(i%2==0) res[0].push_back(s[i]);
                else res[1].push_back(s[i]);
            }
        }
```

```

        ans+=res[0];
        ans+=res[1];
        return ans;
    }
    int flag=0,cout1=0,cout2=numRows-2;
    for(int i=0;i<s.size();i++){
        if(!flag){
            res[cout1++].push_back(s[i]);
            if(cout1==numRows){
                flag=1;
                cout1=0;
            }
        }else if(flag){
            res[cout2--].push_back(s[i]);
            if(!cout2){
                flag=0;
                cout2=numRows-2;
            }
        }
    }

    //string ans;
    for(int i=0;i<numRows;i++){
        ans+=res[i];
    }

    return ans;
}
};

```

```

class Solution {
public:
    string convert(string s, int numRows) {

        if (numRows == 1) return s;

        vector<string> rows(min(numRows, int(s.size()))); //长度的考虑
        int curRow = 0;
        bool goingDown = false;

        //for循环的推荐表达
        for (char c : s) {
            rows[curRow] += c;
            if (curRow == 0 || curRow == numRows - 1) goingDown =
!goingDown;
            //两种情况下问好表达式的妙用。（问好表达式的用法主要式再一种判断两种结果
            的情况下来简化代码）
            curRow += goingDown ? 1 : -1;
        }
    }
}

```

```
        string ret;  
        for (string row : rows) ret += row;  
        return ret;  
    }  
};
```

作者: LeetCode

链接: <https://leetcode-cn.com/problems/zigzag-conversion/solution/z-zi-xing-bian-huan-by-leetcode/>

来源: 力扣 (LeetCode)

著作权归作者所有。商业转载请联系作者获得授权，非商业转载请注明出处。

感悟：简单的题目要学会用精炼的代码给写出来