# 题目

请实现一个函数，把字符串 s 中的每个空格替换成"%20"。

**示例 1：**

输入：s = "We are happy."

输出："We%20are%20happy."

**限制：**

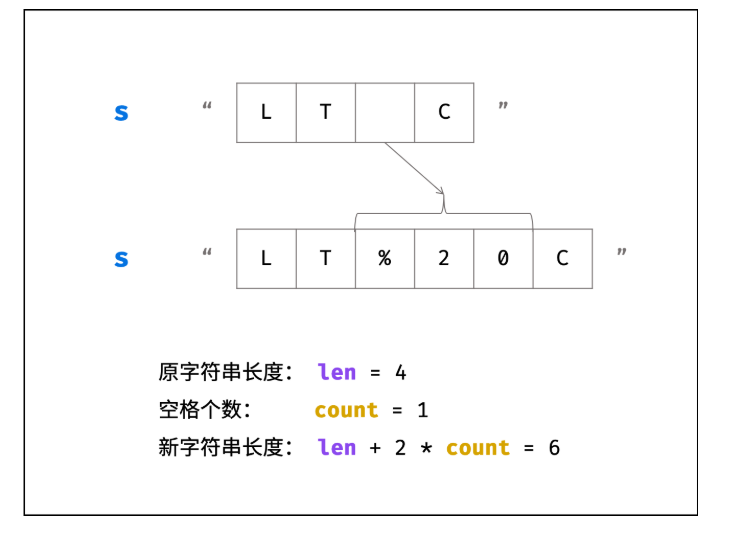
0 <= s 的长度 <= 10000

# 分析

## 方法一：字符数组

在C++语言中，string被设计成「可变」的类型，因此可以在不新建字符串的情况下实现原地修改。

由于需要将空格替换为"%20"，字符串的总字符数增加，因此需要扩展原字符串s的长度，计算公式为：新字符串长度=原字符串长度+ 2 \*空格个数，示例如下图所示。



算法流程：

1、初始化：空格数量count ，字符串s的长度len ；

2、统计空格数量：遍历s ，遇空格则count++；

3、修改s长度：添加完"%20"后的字符串长度应为len + 2 \* count；

4、倒序遍历修改：i指向原字符串尾部元素， j指向新字符串尾部元素；当i = j时跳出（代表左方已没有空格，无需继续遍历）；

当s[i]不为空格时：执行s[j] = s[i]；

当s[i]为空格时：将字符串闭区间[j-2, j] 的元素修改为"%20" ；由于修改了3个元素，因此需要j -= 2；

5、返回值：已修改的字符串s；

**复杂度分析：**

时间复杂度O(N)：遍历统计、遍历修改皆使用O(N)时间。

空间复杂度O(1)：由于是原地扩展s长度，因此使用O(1)额外空间。

**代码：**

class Solution {

public:

string replaceSpace(string s) {

int count = 0, len = s.size();

// 统计空格数量

for (char c : s) {

if (c == ' ') count++;

}

// 修改s长度

s.resize(len + 2 \* count);

// 倒序遍历修改

for(int i = len - 1, j = s.size() - 1; i < j; i--, j--) {

if (s[i] != ' ')

s[j] = s[i];

else {

s[j - 2] = '%';

s[j - 1] = '2';

s[j] = '0';

j -= 2;

}

}

return s;

}

};

版本二：

class Solution {

public:

    string replaceSpace(string s) {

        int space\_cnt=0,oldSize=s.size();

        for(int i=0;i<oldSize;i++)

        {

            if(s.at(i) == ' ')

                space\_cnt++;

        }

        s.resize(oldSize+2\*space\_cnt);

        int newSize=s.size();

        for(int i=oldSize-1,j=newSize-1;i<j;i--,j--)

        {

            if(s[i] != ' ')

            {

                s[j] = s[i];

            }

            else

            {

                s[j-2] = '%';

                s[j-1] = '2';

                s[j] = '0';

                j-=2;

            }

        }

        return s;

    }

};

## 方法二：双指针