# 题目

给定一组字符，使用原地算法将其压缩。

压缩后的长度必须始终小于或等于原数组长度。

数组的每个元素应该是长度为1 的字符（不是 int 整数类型）。

在完成原地修改输入数组后，返回数组的新长度。

**进阶：**

你能否仅使用O(1) 空间解决问题？

**示例 1：**

输入：

["a","a","b","b","c","c","c"]

输出：

返回 6 ，输入数组的前 6 个字符应该是：["a","2","b","2","c","3"]

说明：

"aa" 被 "a2" 替代。"bb" 被 "b2" 替代。"ccc" 被 "c3" 替代。

**示例 2：**

输入：

["a"]

输出：

返回 1 ，输入数组的前 1 个字符应该是：["a"]

解释：

没有任何字符串被替代。

**示例 3：**

输入：

["a","b","b","b","b","b","b","b","b","b","b","b","b"]

输出：

返回 4 ，输入数组的前4个字符应该是：["a","b","1","2"]。

解释：

由于字符 "a" 不重复，所以不会被压缩。"bbbbbbbbbbbb" 被 “b12” 替代。

注意每个数字在数组中都有它自己的位置。

**提示：**

所有字符都有一个ASCII值在[35, 126]区间内。

1 <= len(chars) <= 1000。

# 分析

class Solution {

public:

int compress(vector<char>& chars) {

int size = chars.size();

int cur = 0;

for(int i=0,j=0;i<size;j=i)

{

while(i<size && chars[i]==chars[j])

i++;

chars[cur++] = chars[j];

if(i-j==1)

continue;

string s = to\_string(i-j);

for(int k=0;k<s.size();k++)

chars[cur++] = s[k];

}

return cur;

}

};