# 题目

给你一个仅由 大写 英文字符组成的字符串 s 。

你可以对此字符串执行一些操作，在每一步操作中，你可以从 s 中删除 任一个 "AB" 或 "CD" 子字符串。

通过执行操作，删除所有 "AB" 和 "CD" 子串，返回可获得的最终字符串的最小可能长度。

注意，删除子串后，重新连接出的字符串可能会产生新的 "AB" 或 "CD" 子串。

示例 1：

输入：s = "ABFCACDB"

输出：2

解释：你可以执行下述操作：

- 从 "ABFCACDB" 中删除子串 "AB"，得到 s = "FCACDB" 。

- 从 "FCACDB" 中删除子串 "CD"，得到 s = "FCAB" 。

- 从 "FCAB" 中删除子串 "AB"，得到 s = "FC" 。

最终字符串的长度为 2 。

可以证明 2 是可获得的最小长度。

示例 2：

输入：s = "ACBBD"

输出：5

解释：无法执行操作，字符串长度不变。

提示：

1 <= s.length <= 100

s 仅由大写英文字母组成

# 分析

## 方法一：栈

class Solution {

public:

int minLength(string s) {

vector<int> vec;

for(auto c : s) {

vec.push\_back(c);

int m = vec.size();

if (m >= 2 &&

((vec[m-2] == 'A' && vec[m-1] == 'B') || (vec[m-2] == 'C' && vec[m-1] == 'D'))) {

vec.pop\_back();

vec.pop\_back();

}

}

return vec.size();

}

};