# 题目

字符串有三种编辑操作:插入一个英文字符、删除一个英文字符或者替换一个英文字符。 给定两个字符串，编写一个函数判定它们是否只需要一次(或者零次)编辑。

示例 1:

输入:

first = "pale"

second = "ple"

输出: True

示例 2:

输入:

first = "pales"

second = "pal"

输出: False

# 分析

## 方法一：模拟

思路：

这个问题可以通过比较两个字符串的长度以及它们的字符来判断它们是否只需要一次(或者零次)编辑。具体步骤如下：

1、如果两个字符串相等，则它们不需要任何编辑，直接返回 true。

2、如果两个字符串的长度差超过 1，则它们至少需要两次编辑，直接返回 false。

3、如果两个字符串的长度相差为 1，则需要判断它们是否只需要一次插入或删除操作。

4、如果两个字符串的长度相同，则需要判断它们是否只需要一次替换操作。

代码：

class Solution {

public:

bool oneEditAway(string first, string second) {

int m = first.size();

int n = second.size();

// 如果两个字符串相等，则不需要任何编辑

if (first == second) {

return true;

}

// 如果两个字符串的长度差超过 1，则至少需要两次编辑

if (abs(m - n) > 1) {

return false;

}

// 统计不同字符的数量

int diffCount = 0;

if (m == n) {

// 长度相同，判断是否只需要一次替换操作

for (int i = 0; i < m; ++i) {

if (first[i] != second[i]) {

diffCount++;

}

if (diffCount > 1) {

return false;

}

}

} else {

// 长度不同，判断是否只需要一次插入或删除操作

string& longer = m > n ? first : second;

string& shorter = m > n ? second : first;

int i = 0, j = 0;

while (i < longer.size() && j < shorter.size()) {

if (longer[i] != shorter[j]) {

diffCount++;

if (diffCount > 1) {

return false;

}

i++;

} else {

i++;

j++;

}

}

}

return true;

}

};

在这个实现中，我们首先判断两个字符串是否相等，如果相等，则不需要任何编辑，直接返回 true。然后，我们判断两个字符串的长度差是否超过 1，如果超过，则至少需要两次编辑，直接返回 false。接着，我们分别处理长度相同和长度不同的情况。对于长度相同的情况，我们统计不同字符的数量，如果不同字符的数量超过 1，则返回 false。对于长度不同的情况，我们将较长的字符串作为长字符串，较短的字符串作为短字符串，然后逐个比较字符，统计不同字符的数量，如果不同字符的数量超过 1，则返回 false。如果遍历结束后仍然没有返回 false，则说明两个字符串只需要一次编辑，返回 true。