# 题目

给定两个字符串 a 和 b，寻找重复叠加字符串 a 的最小次数，使得字符串 b 成为叠加后的字符串 a 的子串，如果不存在则返回 -1。

注意：字符串 "abc" 重复叠加 0 次是 ""，重复叠加 1 次是 "abc"，重复叠加 2 次是 "abcabc"。

示例 1：

输入：a = "abcd", b = "cdabcdab"

输出：3

解释：a 重复叠加三遍后为 "abcdabcdabcd", 此时 b 是其子串。

示例 2：

输入：a = "a", b = "aa"

输出：2

示例 3：

输入：a = "a", b = "a"

输出：1

示例 4：

输入：a = "abc", b = "wxyz"

输出：-1

提示：

1 <= a.length <= 104

1 <= b.length <= 104

a 和 b 由小写英文字母组成

# 分析

## 方法一：暴力法

思路：

代码：

class Solution {

public:

int repeatedStringMatch(string a, string b) {

int count = 1;

string repeated = a;

// 不断重复叠加字符串 a，直到字符串 b 成为其子串或者重复叠加次数超过 2 \* a.length() + b.length()

while (repeated.find(b) == string::npos && repeated.length() <= 2 \* a.length() + b.length()) {

// 重复叠加一次

repeated += a;

count++;

}

// 如果超过 2 \* a.length() + b.length() 仍然没有找到子串，则返回 -1

if (repeated.find(b) == string::npos) {

return -1;

}

return count;

}

};

## 方法二：Rabin-Karp算法

## 方法三：KMP算法