# Problem 1055: Shortest Way to Form String

## Problem Information

**Difficulty:** Medium
**Acceptance Rate:** 61.44%
**Paid Only:** Yes
**Tags:** Two Pointers, String, Binary Search, Greedy

## Problem Description

A **subsequence** of a string is a new string that is formed from the original string by deleting some (can be none) of the characters without disturbing the relative positions of the remaining characters. (i.e., `"ace"` is a subsequence of `"_a_ b _c_ d _e_ "` while `"aec"` is not).

Given two strings `source` and `target`, return _the minimum number of**subsequences** of _`source` _such that their concatenation equals_`target`. If the task is impossible, return `-1`.

**Example 1:**

**Input:** source = "abc", target = "abcbc" **Output:** 2 **Explanation:** The target "abcbc" can be formed by "abc" and "bc", which are subsequences of source "abc".

**Example 2:**

**Input:** source = "abc", target = "acdbc" **Output:** -1 **Explanation:** The target string cannot be constructed from the subsequences of source string due to the character "d" in target string.

**Example 3:**

**Input:** source = "xyz", target = "xzyxz" **Output:** 3 **Explanation:** The target string can be constructed as follows "xz" + "y" + "xz".

**Constraints:**

* `1 <= source.length, target.length <= 1000` * `source` and `target` consist of lowercase English letters.

## Code Snippets

**C++:**

```
class Solution {
public:
int shortestWay(string source, string target) {

}
};
```

**Java:**

```
class Solution {
public int shortestWay(String source, String target) {

}
}
```

**Python3:**

```
class Solution:
def shortestWay(self, source: str, target: str) -> int:
```