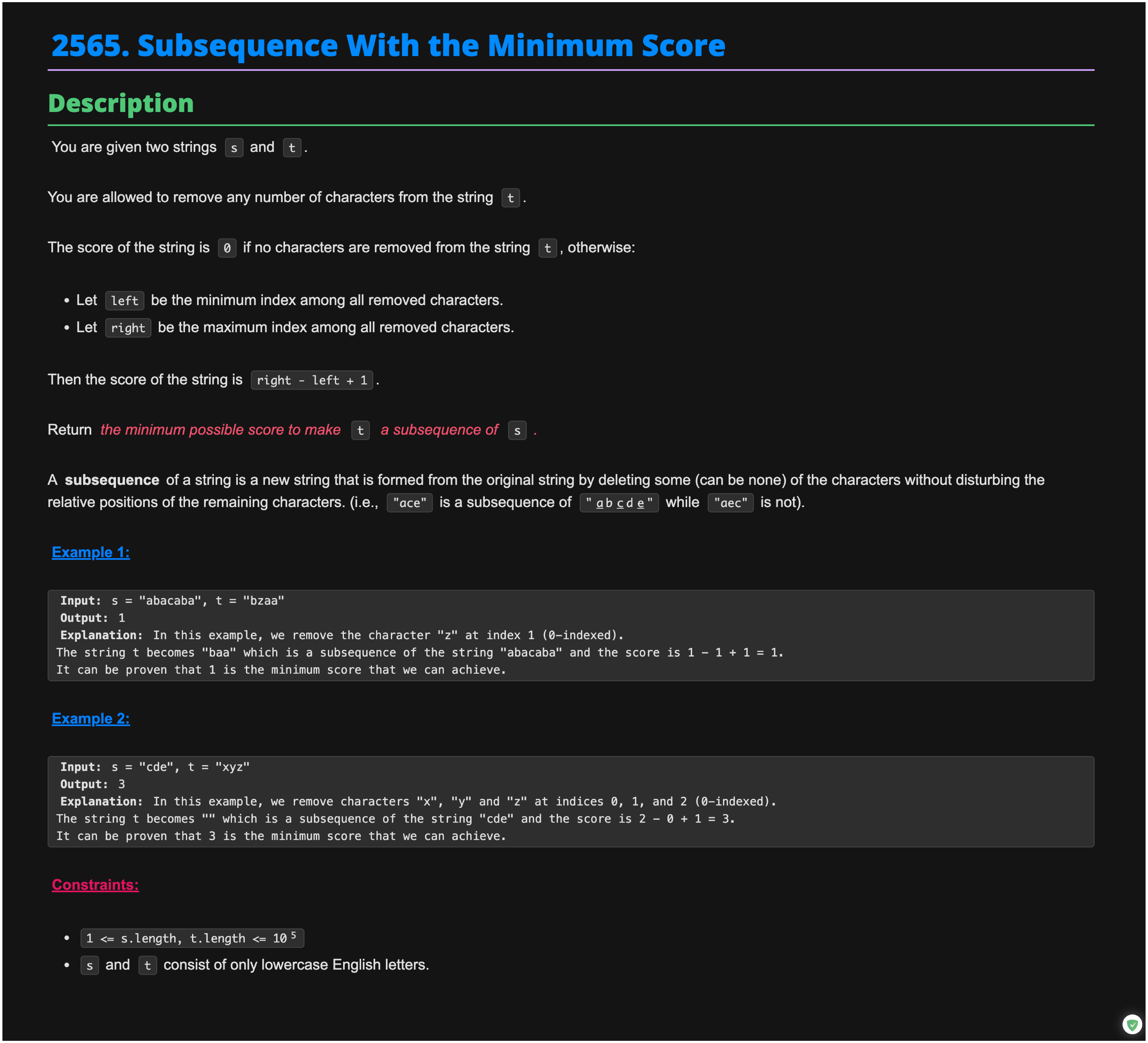

2565. Subsequence With the Minimum Score

Description

You are given two strings `s` and `t`.

You are allowed to remove any number of characters from the string `t`.

The score of the string is `0` if no characters are removed from the string `t`, otherwise:

- Let `left` be the minimum index among all removed characters.
- Let `right` be the maximum index among all removed characters.

Then the score of the string is `right - left + 1`.

Return *the minimum possible score to make* `t` *a subsequence of* `s` *.*

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 `"abcde"` while `"aec"` is not).

Example 1:

```
Input: s = "abacaba", t = "bzaa"
Output: 1
Explanation: In this example, we remove the character "z" at index 1 (0-indexed).
The string t becomes "baa" which is a subsequence of the string "abacaba" and the score is 1 - 1 + 1 = 1.
It can be proven that 1 is the minimum score that we can achieve.
```

Example 2:

```
Input: s = "cde", t = "xyz"
Output: 3
Explanation: In this example, we remove characters "x", "y" and "z" at indices 0, 1, and 2 (0-indexed).
The string t becomes "" which is a subsequence of the string "cde" and the score is 2 - 0 + 1 = 3.
It can be proven that 3 is the minimum score that we can achieve.
```

Constraints:

- $1 <= s.length, t.length <= 10^5$
- `s` and `t` consist of only lowercase English letters.