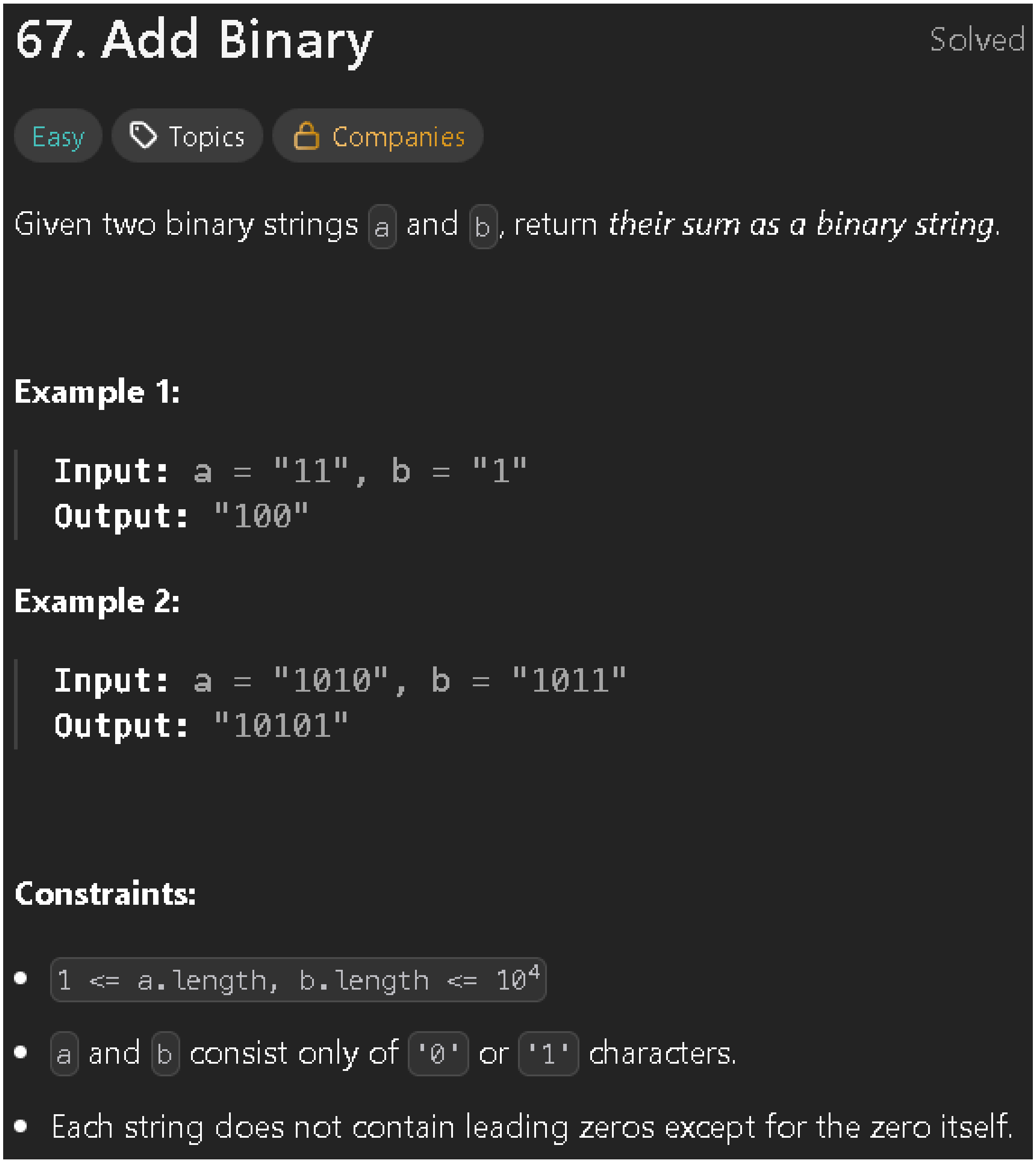

67. Add Binary

Given two binary strings `a` and `b`, return *their sum as a binary string*.

Example 1:

```
Input: a = "11", b = "1"
Output: "100"
```

Example 2:

```
Input: a = "1010", b = "1011"
Output: "10101"
```

Constraints:

- `1 <= a.length, b.length <=` 10^4
- `a` and `b` consist only of `'0'` or `'1'` characters.
- Each string does not contain leading zeros except for the zero itself.

Python:

```
class Solution:
    def addBinary(self, a: str, b: str) -> str:
        return bin(int(a, 2) + int(b, 2))[2:]
        # result = []
        # carry = 0

        # i, j = len(a) - 1, len(b) - 1

        # while i >= 0 or j >= 0 or carry:
```

```
        #    total = carry
        #    if i >= 0:
        #        total += int(a[i])
        #        i -= 1
        #    if j >= 0:
        #        total += int(b[j])
        #        j -= 1

        #    result.append(str(total % 2))  # store the current bit
        #    carry = total // 2             # update carry

        # return ''.join(reversed(result))
```

JavaScript:

```
/**
 * @param {string} a
 * @param {string} b
 * @return {string}
 */
var addBinary = function(a, b) {
    let i = a.length - 1;
    let j = b.length - 1;
    let carry = 0;
    let result = [];

    while (i >= 0 || j >= 0 || carry > 0) {
        let sum = carry;

        if (i >= 0) sum += Number(a[i--]); // add digit from a
        if (j >= 0) sum += Number(b[j--]); // add digit from b

        result.push(sum % 2);   // current bit
        carry = Math.floor(sum / 2); // update carry
    }

    return result.reverse().join('');
};
```

Java:

```
class Solution {
    public String addBinary(String a, String b) {
        StringBuilder result = new StringBuilder();

        int i = a.length() - 1;
        int j = b.length() - 1;
        int carry = 0;

        // Traverse both strings from right to left
        while (i >= 0 || j >= 0 || carry > 0) {
            int sum = carry;

            if (i >= 0) {
                sum += a.charAt(i) - '0'; // convert char to int
                i--;
            }
            if (j >= 0) {
                sum += b.charAt(j) - '0';
                j--;
            }

            // Append current bit
            result.append(sum % 2);

            // Update carry
            carry = sum / 2;
        }

        // Reverse result because we added from right to left
        return result.reverse().toString();
    }
}
```