Problem 3666: Minimum Operations to Equalize Binary String

Problem Information

Difficulty: Hard
Acceptance Rate: 19.08%
Paid Only: No
Tags: Math, String, Breadth-First Search, Union Find, Ordered Set

Problem Description

You are given a binary string `s`, and an integer `k`.

In one operation, you must choose **exactly** `k` **different** indices and **flip** each `'0'` to `'1'` and each `'1'` to `'0'`.

Return the **minimum** number of operations required to make all characters in the string equal to `'1'`. If it is not possible, return -1.

Example 1:

Input: s = "110", k = 1

Output: 1

Explanation:

* There is one `'0'` in `s`. * Since `k = 1`, we can flip it directly in one operation.

Example 2:

Input: s = "0101", k = 3

Output: 2

Explanation:

One optimal set of operations choosing `k = 3` indices in each operation is:

* **Operation 1** : Flip indices `[0, 1, 3]`. `s` changes from `"0101"` to `"1000"`. * **Operation 2** : Flip indices `[1, 2, 3]`. `s` changes from `"1000"` to `"1111"`.

Thus, the minimum number of operations is 2.

Example 3:

Input: s = "101", k = 2

Output: -1

Explanation:

Since `k = 2` and `s` has only one `'0'`, it is impossible to flip exactly `k` indices to make all `'1'`. Hence, the answer is -1.

Constraints:

* `1 <= s.length <= 10■■■■■■■5` * `s[i]` is either `'0'` or `'1'`. * `1 <= k <= s.length`

Code Snippets

C++:

```
class Solution {
public:
int minOperations(string s, int k) {

}
};
```

Java:

```
class Solution {
public int minOperations(String s, int k) {
```

```
}
}
```

Python3:

```
class Solution:
def minOperations(self, s: str, k: int) -> int:
```