

1529. Minimum Suffix Flips

Description

You are given a **0-indexed** binary string `target` of length `n`. You have another binary string `s` of length `n` that is initially set to all zeros. You want to make `s` equal to `target`.

In one operation, you can pick an index `i` where $0 \leq i < n$ and flip all bits in the **inclusive** range `[i, n - 1]`. Flip means changing `'0'` to `'1'` and `'1'` to `'0'`.

Return *the minimum number of operations needed to make* `s` *equal to* `target`.

Example 1:

```
Input: target = "10111"
Output: 3
Explanation: Initially, s = "00000".
Choose index i = 2: "00 000" -> "00 111"
Choose index i = 0: "00111" -> "11000"
Choose index i = 1: "1 1000" -> "1 0111"
We need at least 3 flip operations to form target.
```

Example 2:

```
Input: target = "101"
Output: 3
Explanation: Initially, s = "000".
Choose index i = 0: "000" -> "111"
Choose index i = 1: "1 11" -> "1 00"
Choose index i = 2: "10 0" -> "10 1"
We need at least 3 flip operations to form target.
```

Example 3:

```
Input: target = "00000"
Output: 0
Explanation: We do not need any operations since the initial s already equals target.
```

Constraints:

- `n == target.length`
- $1 \leq n \leq 10^5$
- `target[i]` is either `'0'` or `'1'`.

