424. Longest Repeating Character Replacement

Created	@February 3, 2022 11:13 AM
Difficulty	Medium
LC Url	https://leetcode.com/problems/longest-repeating-character-replacement/
Importance	****
Tag	NEET Sliding Window String
Video	https://www.youtube.com/watch?v=gqXU1UyA8pk

You are given a string `s` and an integer `k`. You can choose any character of the string and change it to any other uppercase English character. You can perform this operation at most `k` times.

Return *the length of the longest substring containing the same letter you can get after performing the above operations*.

Example 1:

```
Input: s = "ABAB", k = 2
Output: 4
Explanation: Replace the two 'A's with two 'B's or vice versa.
```

Example 2:

```
Input: s = "AABABBA", k = 1
Output: 4
Explanation: Replace the one 'A' in the middle with 'B' and form "AABBBBA".
The substring "BBBB" has the longest repeating letters, which is 4.
```

Constraints:

- `1 <= s.length <= 10 5`
- `s` consists of only uppercase English letters.
- `0 <= k <= s.length`

Solution

O(n)

```
class Solution:
    def characterReplacement(self, s: str, k: int) -> int:
        count = {}
        res = 0

        left = 0
        max_freq = 0

        for right in range(len(s)):
            count[s[right]] = count.get(s[right], 0) + 1
            max_freq = max(max_freq, count[s[right]])

            while (right - left + 1) - max_freq > k:
                count[s[left]] -= 1
                left += 1

            res = max(res, right - left + 1)

        return res
```