Problem Statement

You're given:

- A string s (only uppercase English letters).
- An integer k.

You can choose **at most k characters** in the string and replace them with any letter.

👉 Goal: Find the **length of the longest substring** you can get where **all characters are the same** after replacements.

Example

s = "AABABBA", k = 1
Answer: **4**

Because:

- Replace the last B → "AABAAA" → longest substring of "AAAA" has length 4.

Approach (Sliding Window)

1. Use a **sliding window** to keep track of a valid substring.
2. Track the **frequency of the most common character** inside the window.
3. Window is valid if:
   (window length - most frequent char count) <= k
4.
   (Because these are the characters we'd need to replace).
5. Expand the window by moving right.
   If invalid, shrink from left.
6. Keep track of the max length.

Java Solution

```
class Solution {
   public int characterReplacement(String s, int k) {
      int[] freq = new int[26];
      int left = 0, maxCount = 0, maxLen = 0;

      for (int right = 0; right < s.length(); right++) {
         int idx = s.charAt(right) - 'A';
         freq[idx]++;
         maxCount = Math.max(maxCount, freq[idx]);

         // if too many replacements needed, shrink window
         while ((right - left + 1) - maxCount > k) {
            freq[s.charAt(left) - 'A']--;
            left++;
         }

         maxLen = Math.max(maxLen, right - left + 1);
      }
      return maxLen;
   }
}
```

Dry Run (Example: s = "AABABBA", k = 1)

left	right	Window	Freq(A,B)	maxCount	WindowLen	ReplaceNeeded	Valid?	maxLen
0	0	A	A=1,B=0	1	1	0	✅	1
0	1	AA	A=2,B=0	2	2	0	✅	2
0	2	AAB	A=2,B=1	2	3	1	✅	3
0	3	AABA	A=3,B=1	3	4	1	✅	4
0	4	AABAB	A=3,B=2	3	5	2 (>k)	❌	shrink
1	4	ABAB	A=2,B=2	2	4	2 (>k)	❌	shrink
2	4	BAB	A=1,B=2	2	3	1	✅	4
2	5	BABB	A=1,B=3	3	4	1	✅	4
2	6	BABBA	A=2,B=3	3	5	2 (>k)	❌	shrink
3	6	ABBA	A=2,B=2	2	4	2 (>k)	❌	shrink

4	6	BBA	A=1,B=2	2	3	1	✅	4

✅ Final Answer = **4**

Learning Points

1. **Sliding window** avoids O(n^2) brute force.
2. **maxCount** ensures we only care about the most frequent character.
3. Condition (window size - maxCount > k) tells us if we exceeded allowed replacements.