424. Longest Repeating Character Replacement (Medium)

You are given a string `s` consisting of only uppercase english characters and an integer `k`. You can choose up to `k` characters of the string and replace them with any other uppercase English character.

After performing at most `k` replacements, return the length of the longest substring which contains only one distinct character.

Example 1:

```
Input: s = "XYYX", k = 2

Output: 4
```

Explanation: Either replace the 'X's with 'Y's, or replace the 'Y's with 'X's.

Example 2:

```
Input: s = "AAABABB", k = 1

Output: 5
```

Constraints:

- `1 <= s.length <= 1000`
- `0 <= k <= s.length`

▼ 思路:

作法一：hashset 儲存記錄過字母，針對每個字母對整串字串以sliding window搜尋

T:$O(n*m)$, S:$O(m)$ → n為s長度，m出現過字母

作法二：hashmap 儲存記錄過字母，迭代更新最大window

作法一

```
class Solution {
public:
    int characterReplacement(string s, int k) {
        int ans=0;
        unordered_set<char> check(s.begin(),s.end());
        for(char c: check){
            int l=0, count=0;
            for(int r=0;r<s.size();r++){
                if(s[r] == c){
                    count++;
                }
                while(r-l+1 - count > k){
                    if(s[l] == c){
                        count--;
                    }
                    l++;
                }
                ans = (ans > r-l+1) ? ans : r-l+1;
            }
        }
        return ans;
    }
};
```

作法二

```
class Solution {
public:
    int characterReplacement(string s, int k) {
        int ans=0,l=0,maxw=0;
        unordered_map<char,int> check;
        for(int r=0;r<s.size();r++){
            check[s[r]]++;
            maxw = (maxw > check[s[r]]) ? maxw :check[s[r]]; //max windows
```

```
            while(r-l+1 - maxw > k){ // only larger than the max window then do
                check[s[l]]--;
                l++;
            }
            ans = (ans>r-l+1)? ans : r-l+1;
        }
        return ans;
    }
};
```