424. Longest Repeating Character Replacement

Attempted ◎

`Medium`   🏷 Topics   🔒 Companies

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
```

Main problem is the facing is that ... how many character are change ...and ... which have to change....

_____
_____

==*max character is change is = maxlength -maximum frequency of element* ...==

_____

Brute force

```java
1   class Solution {
2       public int characterReplacement(String s, int k) {
3
4           int maxl=0 ,n=s.length() ;
5           for(int i=0;i<n;i++){
6               int maxf=0;
7                int[] arr=new int[26];
8
9               for(int j=i;j<n;j++){
10                  arr[s.charAt(j)-'A']++;
11                  maxf=Math.max(maxf , arr[s.charAt(j)-'A']);
12                  if(k>=(j-i+1-maxf)){
13                      maxl=Math.max(maxl , j-i+1);
14                  }else{
15                      break;
16                  }
17              }
18
19          }
20
21          return maxl;
22      }
23  }
```

Better sol:

```java
class Solution {
    public int characterReplacement(String s, int k) {
        int[] arr=new int[26];
        int maxl=0 , l=0 , r=0 , maxf=0 , n=s.length();
        while(r<n){
            arr[s.charAt(r)-'A']++;
            maxf=Math.max(maxf , arr[s.charAt(r)-'A']);
            while(k<(r-l+1-maxf)){
                arr[s.charAt(l)-'A']--;
                 maxf=0;
                for(int i=0 ;i<26;i++) maxf =Math.max(maxf,arr[i]);
                l++;
            }
            if(k>=(r-l+1-maxf)){
                maxl=Math.max(maxl , r-l+1);
            }
            r++;
        }
        return maxl;
    }
}
```

TC -O(N+N)*26;

12 March 2025   02:13

Optimal Approach

	Tc -O(N)
| ```java
class Solution {
    public int characterReplacement(String s, int k) {
        int[] arr=new int[26];
        int maxl=0 , l=0 , r=0 , maxf=0 , n=s.length();
        while(r<n){
            arr[s.charAt(r)-'A']++;
            maxf=Math.max(maxf , arr[s.charAt(r)-'A']);
            if(k<(r-l+1-maxf)){
                arr[s.charAt(l)-'A']--;
                l++;
            }
            if(k>=(r-l+1-maxf)){
                maxl=Math.max(maxl , r-l+1);
            }
            r++;
        }
        return maxl;
    }
}
``` | |