1371. Find the Longest Substring Containing Vowels in Even Counts

Given the string `s`, return the size of the longest substring containing each vowel an even number of times. That is, 'a', 'e', 'i', 'o', and 'u' must appear an even number of times.

Example 1:

```
Input: s = "eleetminicoworoep"
Output: 13
Explanation: The longest substring is "leetminicowor" which contains two each of the vowels: e, i and o and zero of the vowels: a and u.
```

Example 2:

```
Input: s = "leetcodeisgreat"
Output: 5
Explanation: The longest substring is "leetc" which contains two e's.
```

Example 3:

```
Input: s = "bcbcbc"
Output: 6
Explanation: In this case, the given string "bcbcbc" is the longest because all vowels: a, e, i, o and u appear zero times.
```

Constraints:

- $1 <= s.length <= 5 \times 10^5$
- `s` contains only lowercase English letters.

1371. Find the Longest Substring Containing Vowels in Even Counts

```
/*
   Brute force - TLE
   Time complexity: O(n^2 *(26+n+26)) = O(n^3+2*26.n^2)=O(n^3)
   Space complexity: O(26)=O(1)
*/
class Solution {
   public:
      int findTheLongestSubstring(std::string s) {
         int n=s.size();
         int ans=0;
         for(int left=0;left<n;++left){
            for(int right=left;right<n;++right){
               std::vector<int> freq(26,0);
               for(int i=left;i<=right;++i){
                  if(s[i]=='a' || s[i]=='e' || s[i]=='i' || s[i]=='o' || s[i]=='u') freq[s[i]-'a']++;
               }

               int sum=0;
               for(int i=0;i<26;++i){
                  freq[i]%=2;
                  sum+=freq[i];
               }

               if(sum==0) ans=std::max(ans,right-left+1);
            }
         }
         return ans;
      }
};
```

1371. Find the Longest Substring Containing Vowels in Even Counts

```
/*
   Preprecessing + Brute force - TLE
   Time complexity: O(5n+n+n^2)=O(n^2)
   Space complexity: O(5n)=O(n)
*/
typedef std::vector<int> vi;
typedef std::vector<vi> vvi;
class Solution {
   public:
      int findTheLongestSubstring(std::string s) {
         int n=s.size();

         vvi pre(5,vi(n+1));
         for(int i=1;i<=n;++i){
            pre[0][i]=pre[0][i-1]+int(s[i-1]=='a');
            pre[1][i]=pre[1][i-1]+int(s[i-1]=='e');
            pre[2][i]=pre[2][i-1]+int(s[i-1]=='i');
            pre[3][i]=pre[3][i-1]+int(s[i-1]=='o');
            pre[4][i]=pre[4][i-1]+int(s[i-1]=='u');
         }

         int ans=0;
         for(int left=0;left<n;++left){
            for(int right=left;right<n;++right){
               int sum_a=pre[0][right+1]-pre[0][left];
               int sum_e=pre[1][right+1]-pre[1][left];
               int sum_i=pre[2][right+1]-pre[2][left];
               int sum_o=pre[3][right+1]-pre[3][left];
               int sum_u=pre[4][right+1]-pre[4][left];

               if(sum_a%2==0&&sum_e%2==0&&sum_i%2==0&&sum_o%2==0&&sum_u%2==0)
ans=std::max(ans,right-left+1);
            }
         }
         return ans;
      }
};
```

1371. Find the Longest Substring Containing Vowels in Even Counts

```
/*
   Bitmask version 1
   Time complexity: O(32+n)=O(n)
   Space complexity: O(2^5)=O(32)=O(1)
*/
typedef std::vector<int> vi;
typedef std::vector<vi> vvi;
class Solution {
   public:
      int findTheLongestSubstring(std::string s) {
         int n=s.size();

         vi seen(32,INT_MIN);
         seen[0]=-1;
         int mask=0;
         int ans=0;
         for(int i=0;i<n;++i){
            if(s[i]=='a') mask^=(1<<4);
            else if(s[i]=='e') mask^=(1<<3);
            else if(s[i]=='i') mask^=(1<<2);
            else if(s[i]=='o') mask^=(1<<1);
            else if(s[i]=='u') mask^=(1<<0);
            if(seen[mask]!=INT_MIN) ans=std::max(ans,i-seen[mask]);
            else seen[mask]=i;
         }
         return ans;
      }
};
```

1371. Find the Longest Substring Containing Vowels in Even Counts

```
/*
   Bitmask version 2
   Time complexity: O(32+n)=O(n)
   Space complexity: O(2^5)=O(32)=O(1)
*/
typedef std::vector<int> vi;
typedef std::vector<vi> vvi;
class Solution {
   public:
      int findTheLongestSubstring(std::string s) {
         int n=s.size();

         vi seen(32,INT_MIN);
         seen[0]=-1;
         int mask=0;
         int ans=0;
         for(int i=0;i<n;++i){
            if(s[i]=='a') mask^=16;
            else if(s[i]=='e') mask^=8;
            else if(s[i]=='i') mask^=4;
            else if(s[i]=='o') mask^=2;
            else if(s[i]=='u') mask^=1;
            if(seen[mask]!=INT_MIN) ans=std::max(ans,i-seen[mask]);
            else seen[mask]=i;
         }
         return ans;
      }
};
```