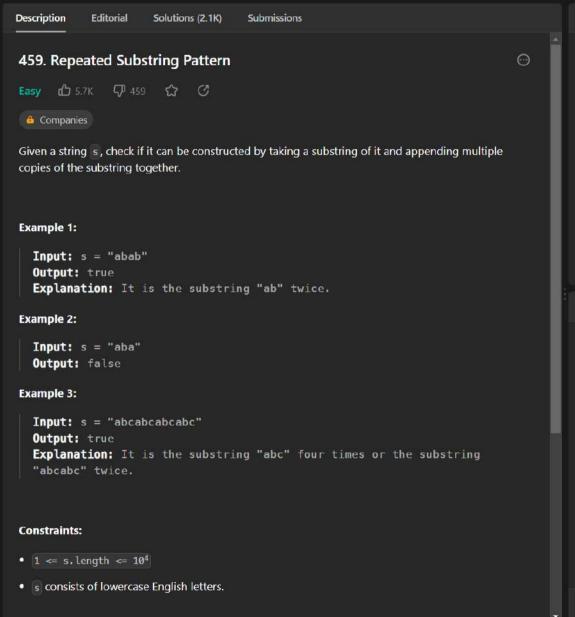


```
i Java ∨ 📗 🖨 Auto
         public int firstUniqChar(String s) {
              int freq[] = new int [26];
             Queue<Character> q = new LinkedList<>();
              for(int i = 0; i < s.length(); i++){</pre>
                 char ch = s.charAt(i);
                  q.add(ch);
                  freq[ch-'a']++;
                  while(!q.isEmpty() && freq[q.peek()-'a'] >1){
                      q.remove();
              return a isEmpty()?-1 · s indexOf(a peek()).
Testcase
         Result
Accepted Runtime: 0 ms

    Case 1
              • Case 2
                           • Case 3
Input
  "leetcode"
Output
Expected
Console v
                                                                    Run
                                                                              Submit
```



```
i Java ∨ 📗 🖨 Auto
     public class Solution {
         public boolean repeatedSubstringPattern(String s) {
              int n = s.length();
             for (int i = 1; i \leftarrow n / 2; i++) {
                  if (n \% i == 0) {
                      String substring = s.substring(0, i);
                      StringBuilder repeated = new StringBuilder();
                      for (int j = 0; j < n / i; j++) {
                          repeated.append(substring);
                      if (repeated.toString().equals(s)) return true;
                                                                              Ln 16. Col 2
Testcase
         Result
Accepted Runtime: 0 ms

    Case 1
              • Case 2
                           • Case 3
Input
  "abab"
Output
  true
Expected
Console v
                                                                     Run
                                                                                Submit
```