Longest K unique characters substring

Given a string you need to print the size of the longest possible substring that has exactly K unique characters. If there is no possible substring then print -1.

Example 1:

```
Input:
S = "aabacbebebe", K = 3
Output: 7
Explanation: "cbebebe" is the longest
substring with K distinct characters.
```

Example 2:

```
Input:
S = "aaaa", K = 2
Output: -1
Explanation: There's no substring with K
distinct characters.
```

Your Task:
You don't need to read input or print anything. Your task is to complete the function **longestKSubstr()** which takes the string S and an integer K as input and returns the length of the longest substring with exactly K distinct characters. If there is no substring with exactly K distinct characters then return -1.

Expected Time Complexity: O(|S|).
Expected Auxiliary Space: O(1).

```
def longestKSubstr(self, s, k):
        # code here
        if k>len(s):
            return -1
        ans = 0
        i = -1
        j = -1
        freq = {}
        while True:
            f1, f2 = False, False
            while i < len(s) - 1:
                f1 = True
```

```
            i = i + 1
            ch = s[i]
            freq[ch] = freq.get(ch, 0) + 1
            if len(freq) < k:
                continue
            elif len(freq)==k:
                ans = max(ans ,(i - j))
            else:
                break

        while j < i:
            f2 = True
            j = j + 1
            ch = s[j]
            freq[ch] = freq.get(ch, 0) - 1
            if freq[ch]==0:
                del freq[ch]
            if len(freq)> k:
                continue
            elif len(freq)==k:
                # ans = max(ans ,(i - j))
                break

        if f1 is False and f2 is False:
            break

    return ans if ans!=0 else -1
```