Given a string, find the length of the longest substring T that contains at most *k* distinct characters.

Example 1:

```
Input: s = "eceba", k = 2
Output: 3
Explanation: T is "ece" which its length is 3.
```

```
class Solution {
    public int lengthOfLongestSubstringKDistinct(String s, int k) {
        int r = 0, l = 0, num = 0, max = 0;
        int[] cs = new int[256];
        while (r < s.length()) {
            cs[s.charAt(r)]++;
            if (cs[s.charAt(r)] == 1) {
                num++;
            }
            while (num > k) {
                cs[s.charAt(l)]--;
                if (cs[s.charAt(l)] == 0) {
                    num--;
                }
                l++;
            }
            max = Math.max(max, r - l + 1);
            r++;
        }
        return max;
    }
}
```

Given a string ***s*** , find the length of the longest substring ***t*** that contains **at most** 2 distinct characters.

Example 1:

```
Input: "eceba"
Output: 3
Explanation: t is "ece" which its length is 3.
```

```
class Solution {
    public int lengthOfLongestSubstringTwoDistinct(String s) {
        int r = 0, l = 0, num = 0, max = 0;
        int[] cs = new int[256];
        while (r < s.length()) {
            cs[s.charAt(r)]++;
            if (cs[s.charAt(r)] == 1) {
                num++;
            }
            while (num > 2) {
                cs[s.charAt(l)]--;
                if (cs[s.charAt(l)] == 0) {
                    num--;
                }
                l++;
            }
            max = Math.max(max, r - l + 1);
            r++;
        }
        return max;
    }
}
```