

Longest Substring with K Uniques

Difficulty: Medium Accuracy: 34.65% Submissions: 180K+ Points: 4

Given a string **s**, you need to print the size of the longest possible substring with exactly **k unique** characters. If no possible substring exists, print -1.

Examples:

Input: s = "aabacbebebe", k = 3

Output: 7

Explanation: "cbebebe" is the longest substring with 3 distinct characters.

Input: s = "aaaa", k = 2

Output: -1

Explanation: There's no substring with 2 distinct characters.

Input: s = "aabaaab", k = 2

Output: 7

Explanation: "aabaaab" is the longest substring with 2 distinct characters.

Sol	<pre>import java.util.*; class Solution { public int longestkSubstr(String s, int k) { if (k > s.length()) return -1; // Edge case: impossible to have k distinct characters int l = 0, r = 0, n = s.length(), maxl = -1; Map<Character, Integer> map = new HashMap<>(); while (r < n) { char ch = s.charAt(r); map.put(ch, map.getOrDefault(ch, 0) + 1); while (map.size() > k) { char leftChar = s.charAt(l); map.put(leftChar, map.get(leftChar) - 1); if (map.get(leftChar) == 0) { map.remove(leftChar); } l++; } // Update maxl only when we have exactly k distinct characters if (map.size() == k) { maxl = Math.max(maxl, r - l + 1); } r++; } return maxl; } }</pre>
-----	--