**[Longest Substring Without Repeating Characters](https://leetcode.com/problems/longest-substring-without-repeating-characters/)**

**import** java.util.HashMap;

**public** **class** LongestSubstringWithoutRepeatingCharacters {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

System.***out***.println(*lengthOfLongestSubstring*("bbbbbbb"));

}

**public** **static** **int** lengthOfLongestSubstring(String s) {

**if**(s == **null** || s.length() == 0) {

**return** 0;

}

HashMap<Character, Integer> map = **new** HashMap<>();

**int** maxLength = 0;

**for**(**int** i = 0 , j = 0 ; i < s.length() ; i++) {

**if**(map.containsKey(s.charAt(i))) {

j = Math.*max*(j, map.get(s.charAt(i)));

}

map.put(s.charAt(i), i + 1);

maxLength = Math.*max*(maxLength, i - j + 1);

}

**return** maxLength;

}

}

Time complexity : O(n) , n is length of given string

Space Complexity : O(n) , n is length of given string