Question: https://leetcode.com/problems/longest-substring-without-repeating-characters/

We need to create a hashMap and store each character’s last occurence position as value to that character key.

So now if the character has already existed before then it’s position+1(updatedStart) will be new starting position if the previous starting position was less than updatedStart.

So like this you get length of each substring with unique characters.

So return the max of them.

Code:  
class Solution {

public int lengthOfLongestSubstring(String s) {

HashMap<Character, Integer> mem = new HashMap<>();

int l=0, max=0;

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

char c = s.charAt(i);

if(mem.containsKey(c)){

l=Math.max(l,mem.get(c)+1);

}

mem.put(c,i);

max = Math.max(max, (i-l+1));

}

return max;

}

}

Github Link :<https://lnkd.in/ecwtJeaz>