Longest substring without repeating Characters

Brute-Force Approach

a b c x y a z b x

→ a
→ ab
→ abc
→ abcx
→ abcxy

Pseudo code:

```
maxLen = -99999;
for i=0 to arr.size()          → n
    string sub = " ";
    hash[255] = {0};

    for j=i to arr.size()      → n
        if (hash[arr[j]] == 1)
            break;
        sub += arr[j]
        hash[arr[j]] = 1;

    len = j - i + 1
    maxLen = max(maxLen, len);
```

Time Complexity

$O(n*n) = O(n^2)$

Optimized Solution:

Sliding window with 2 pointer's:

```
function substring (string s)
    int l=0, r=0
    max = 0;
    hash[256] = -1;
    int n = s.size();
    while (r<n){               → n
        if (hash[s[r]] != -1){
            if (hash[s[r]] >= l){
                l = hash[s[r]] + 1
            }
        len = r-l+1
        max = max(max, len)
        hash[s[r]] = r
        r++;
}
```

Time Complexity

$O(n)$