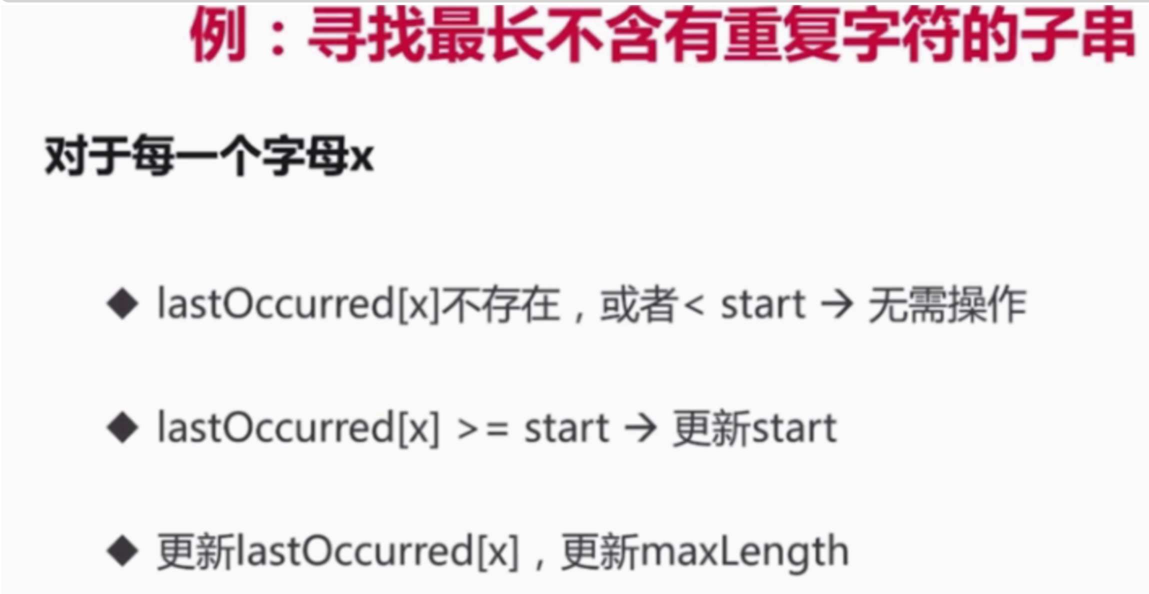
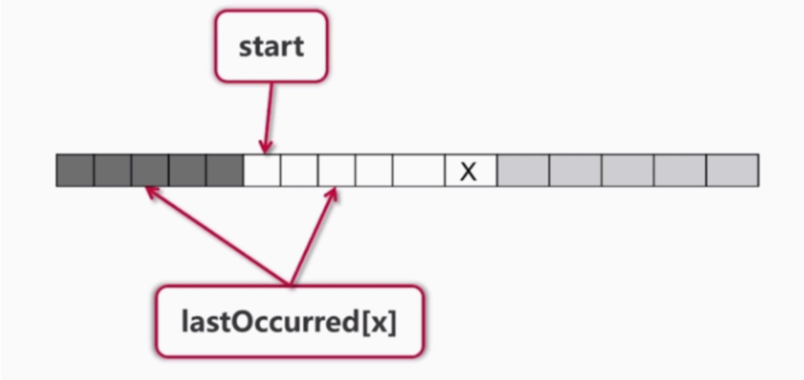
```
// https://leetcode-cn.com/problems/zui-chang-bu-han-zhong-fu-zi-fu-de-zi-zi-fu-chuan-lcof/


func lengthOfLongestSubstring(s string) int {
   lastOccurred := make(map[byte]int)
   start := 0
   maxLength := 0
   for i, ch := range []byte(s) {
      lastI := 0
      ok := false
      if lastI, ok = lastOccurred[ch]; ok && lastI >= start {
         start = lastI + 1
      }
      //fmt.Printf("i = %v, ch = %v, lastI = %v, ok = %v, start = %v, lastOccurred = %v",
      // i, ch, lastI, ok, start, lastOccurred)
      //fmt.Println();


      if i - start + 1 > maxLength {
         maxLength = i - start + 1
      }


      lastOccurred[ch] = i
   }
   return maxLength
}
```

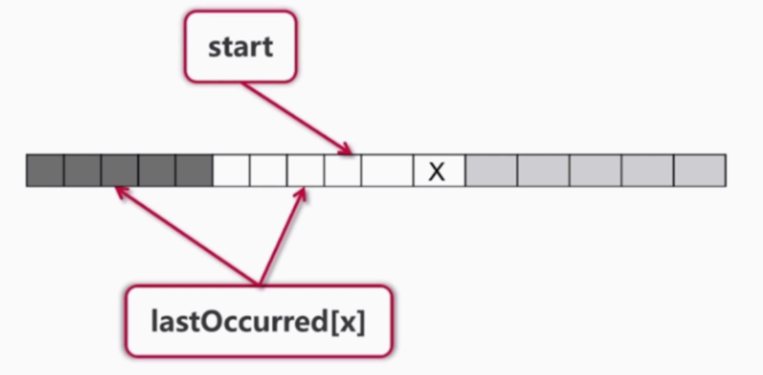


例：寻找最长不含有重复字符的子串

对于每一个字母x

- lastOccurred[x]不存在，或者< start → 无需操作
- lastOccurred[x] >= start → 更新start
- 更新lastOccurred[x]，更新maxLength


start
x
lastOccurred[x]


start
x
lastOccurred[x]